X-551-71-238
PREPRINT

NASA TM X-65620

TBERR TWO-BODY ERROR ANALYSIS PROGRAM

DANIEL P. MUHONEN

MAY 1971

GSFC

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-551-71-238

TBERR

TWO-BODY ERROR ANALYSIS
PROGRAM

Daniel P. Muhonen

May 1971

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

PRECEDING PAGE BLANK NOT FILMED

TWO-BODY ERROR ANALYSIS PROGRAM

Daniel P. Muhonen

ABSTRACT

The Two-Body Error Analysis Program is a rapid computer program designed to evaluate the resulting state vector and covariance matrix for an orbiter after one coast and one burn maneuver. After the burn, histograms are computed for the orbital elements and other state dependent variables, including delta-velocity corrections necessary to attain a prescribed orbit.

The program is coded in FORTRAN IV for the IBM 360/95 computer. This document contains both a user's guide and a complete description of the analysis.

SUMMARY

Initial conditions for this program include a state vector and covariance matrix for an orbiter, together with constant thrust vector and thrusting errors. The program will compute the resulting state vector and covariance matrix after one coast and one burn maneuver. In addition histograms are computed after the burn for orbital elements and other state-dependent variables, including delta-velocity corrections necessary to attain a prescribed orbit. A two-body central force field is assumed throughout. To increase the flexibility of the program, an input option is included which will divide the burn time into a number of sub-intervals. This will reduce numerical truncation errors for long burn maneuvers, and it enables the user to program variable thrust vector and thrusting errors as step functions through the burn. The histograms are printed on the output printer in both graphical and numerical form.

CONTENTS

TABLES

ILLUSTRATIONS

TWO-BODY ERROR ANALYSIS PROGRAM

INTRODUCTION

The two-body error analysis program propagates a state vector and covariance matrix for an orbiter through coast and burn maneuvers. Histograms of the final orbital elements and other state-dependent variables are evaluated by a Monte-Carlo procedure. Provision is included for calculating a histogram of the delta-velocity required to effect prescribed semi-major axis, eccentricity, and inclination (optional).

RESTRICTIONS

Installation	-	GSFC
System and Configuration	-	IBM 360/95
Source Language	-	FORTRAN IV
Subroutines Required	-	MTXPR, GAUSS, RANDU, EIGEN, BURNST, PARTAL, POWERX, MTRPLY, TWOBDY, CONVET, MTRX, TAB1, CROSS, ORB, TCONIC, HISTO, DELVS
Functions Required	-	NSAMP, BARN1, DOT, ARKTNS, FNORM
Storage Required	-	400 K BYTES TOTAL
Common Storage Required	-	None
Input Range and Limitations	-	Monte-Carlo sample size $\leq$ 10,000, number of histogram intervals $\leq$ 50
Output Range, Limitations, and Accuracy	-	Calculations in double precision; two-body force field; linear error propagation; normal distribution of the Cartesian state vector; burn errors uncorrelated.
Running Time	-	0.25 minutes should be allowed per 1000 Monte Carlo sample vectors

INPUTS/OUTPUTS

Remarks

All input is in card form described in Table 1. Initial state vector, covariance matrix, coast time (can be zero), burn time (can be zero), weight, weight loss, thrust vector, thrust errors, and histogram information can be input. A sample input case is given in Figure 1, and the corresponding printer output is included under Figure 2. All output is in units of kilometers, kilograms, seconds, and degrees, unless stated otherwise. Histograms of the following state-dependent parameters, after burn, can be printed:

1. Eccentricity
2. Semi-major axis
3. Inclination
4. Longitude of ascending node
5. Argument of periapsis
6. True anomaly
7. Periapsis radius
8. Apoapsis radius
9. Delta velocity to attain a given semi-major axis, eccentricity, and inclination (optional).

IMPLEMENTATION

Remarks

The program is available on tape with the source and the object on separate files.

I/O Logical Unit Assignments

FORTRAN Logical Unit	Device Type	Use	Space Required
5	Reader	Input	—
6	Printer	Output	

Control Cards or Language

Included with the sample input (see Figure 1)

Table 1

Input Card Formats

Any number of input cases can be stacked; each case must include the following cards:

Card No. 1 . . . Title for output, alphanumeric, columns 1-80.
Card No. 2 . . . Namelist name, &TBDATA, columns 2-8.

The remaining data is in namelist format, i.e., Parameter = ... , ... , ... , columns 2-72. The following parameters can be input; if they are not, either default values or previous stacked-case values are assumed.

Parameter	Dimension	Default Value	Description
DTCI	1	0.	Coast time interval (sec.)
DTBI	1	0.	Total burn time interval (sec.)
NBURN	1	1	Number of computation increments in burn interval (see method).
WI	1	0.	Initial weight (lb.)
WCI	1	0.	Total weight increase during burn (lb.) (generally negative)
PITCHI	1	0.	Initial nominal pitch during burn (deg.)
YAWI	1	0.	Initial nominal yaw during burn (deg.)
THRUSI	1	0.	Nominal thrust during burn (lb.)
ZDELT	1	0.	Uncertainty for Monte-Carlo sample size (used only if $1 \leq \text{NCONF} \leq 6$)
NCONF	1	0	= 0 implies no histograms generated = 1 implies confidence level = .9 = 2 implies confidence level = .95 = 3 implies confidence level = .98

Table 1 (continued)

Parameter	Dimension	Default Value	Description
NCONF (cont.)	1	0	= 4 implies confidence level = .99 = 5 implies confidence level = .995 = 6 implies confidence level = .998 > 6 implies sample size = NCONF
AS	1	0.	= 0 implies no delta-velocity histogram computed > 0 implies delta-velocity histogram computed to attain semi-major axis = AS.
ES	1	0.	Delta-velocity histogram computed to attain eccentricity = ES.
AIS	1	-1.	< 0 implies delta-velocity histogram includes no inclination changes, ≥ 0 implies delta-velocity computed to attain inclination = AIS.
SIGBI	3	0., 0., 0.	Thrust error σ_T (lb.), Pitch error σ_α (deg.), Yaw error σ_ψ (deg.)
IHIST	6	1, 2, 3, 4, 5, 6	Flags to indicate which histograms are desired. Computations are provided for nine state dependent parameters, and up to six of these can be chosen for histograms. The i^{TH} value of IHIST implies that the i^{TH} histogram will be as follows: IHIST (i)

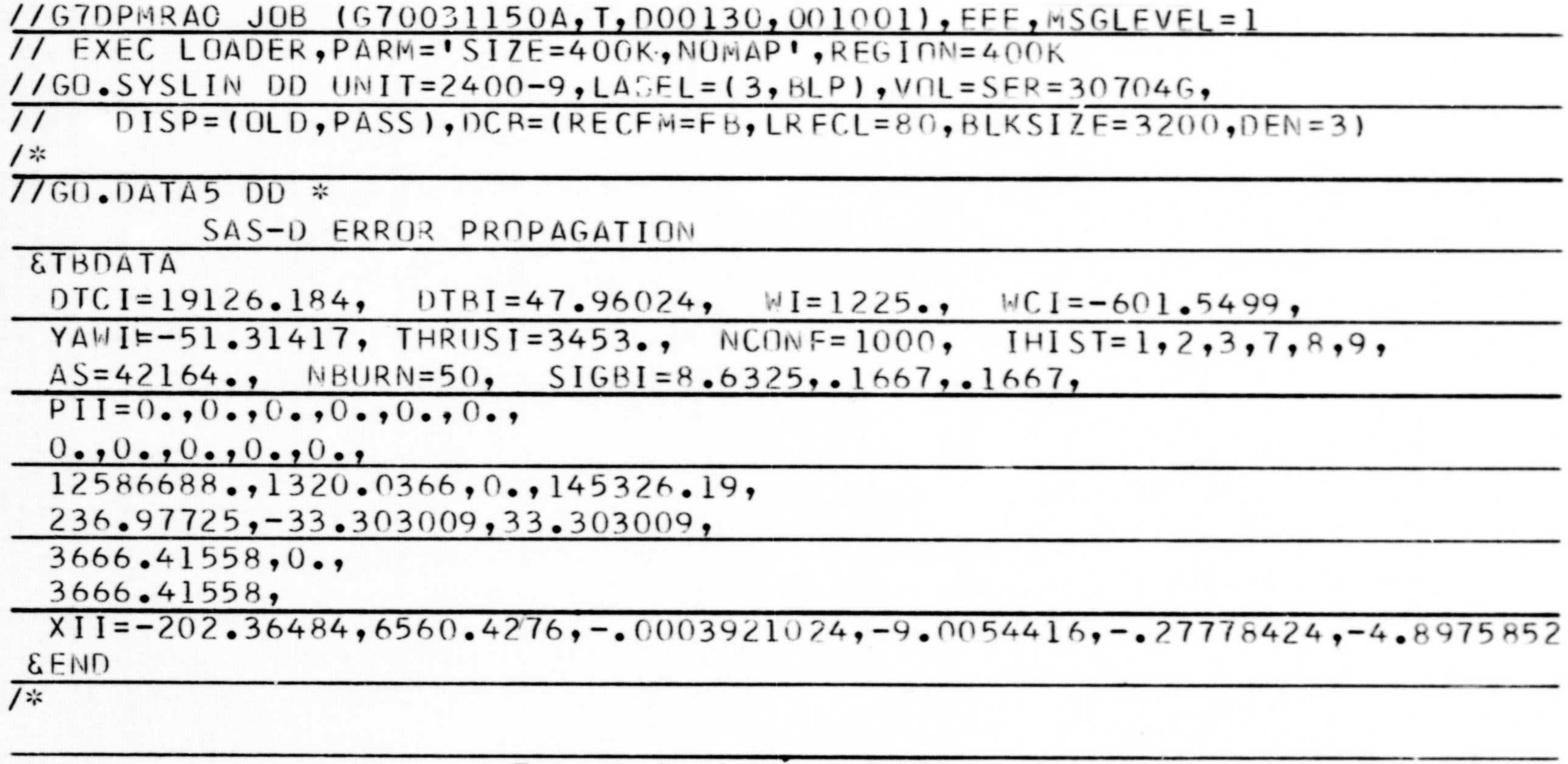
Table 1 (continued)

Parameter	Dimension	Default Value	Description
IHIST (cont.)	6	1, 2, 3, 4, 5, 6	= 0 . . . not printed = 1 . . . eccentricity = 2 . . . semi-major axis = 3 . . . inclination = 4 . . . longitude of ascending node = 5 . . . argument of periapsis = 6 . . . true anomaly = 7 . . . periapsis radius = 8 . . . apoapsis radius = 9 . . . delta velocity
ZUB	(3, 6)	ZUB(1, i) = 0.	ZUB (1, i) = beginning value of i^{TH} histogram
		ZUB(2, i) = 50.	ZUB (2, i) = number of intervals in i^{TH} histogram
		ZUB(3, i) = 0.	ZUB (3, i) = final value of i^{TH} histogram
		i = 1, . . . , 6	[ZUB (1, i) = ZUB (3, i) = 0] implies that the limits for the i^{TH} histogram are those of the sample generated.
XII	6	0.,0.,0.,0.,0.,0.	Initial state vector, $\begin{pmatrix} \vec{r} \\ \dot{\vec{r}} \end{pmatrix}$, in inertial cartesian coordinates (km., km./sec.)
PII	21	0., 0., . . . , 0.	Upper half of initial covariance matrix, input row-wise [e.g., PII (6) = $P_{1,6}$, PII (7) = P_{22}].
IPCOOR	1	1	Coordinates and units flag for PII. = 1 implies local tangent coordinates, $[(\vec{r} \times \dot{\vec{r}}) \times \vec{r},\ \vec{r} \times \dot{\vec{r}},\ \vec{r}\ ]$, and units in ft.^{2} and ft.^{2}/sec.^{2}

Table 1 (continued)

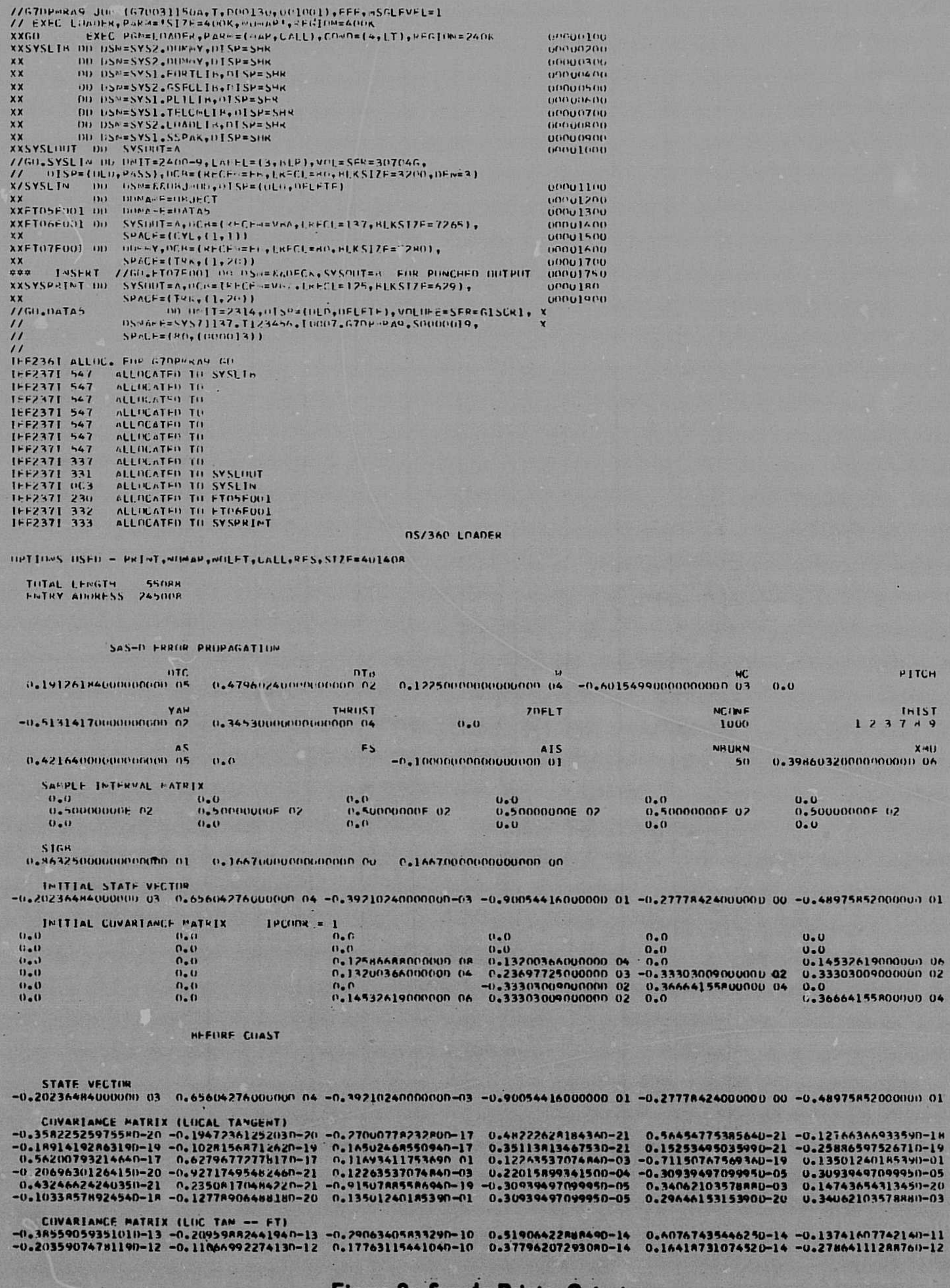
Parameter	Dimension	Default Value	Description
IPCOOR (cont.)	1	1	= 2 implies local tangent coordinates and units in km.^{2} and km.^{2}/sec.^{2}. = 3 implies inertial coordinates of XII and units in ft.^{2} and ft.^{2}/sec.^{2}. = 4 implies inertial coordinates of XII and units in km.^{2} and km.^{2}/sec.^{2}.
XMU	1	398603.2 (μ of earth)	Gravitional constant times mass of central body (km.^{3}/sec.^{2}).

FINAL CARD . . . . namelist data end flag, &END, columns 2-5.



```
//G7DPMRAC JOB (G70031150A,T,D00130,001001),EFF,MSGLEVEL=1
// EXEC LOADER,PARM='SIZE=400K,NOMAP',REGION=400K
//GO.SYSLIN DD UNIT=2400-9,LABEL=(3,BLP),VOL=SER=30704G,
//    DISP=(OLD,PASS),DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DEN=3)
/*
//GO.DATA5 DD *
          SAS-D ERROR PROPAGATION
 &TBDATA
  DTCI=19126.184,  DTBI=47.96024,  WI=1225.,  WCI=-601.5499,
  YAWI=-51.31417, THRUSI=3453.,  NCONF=1000,  IHIST=1,2,3,7,8,9,
  AS=42164.,  NBURN=50,  SIGBI=8.6325,.1667,.1667,
  PII=0.,0.,0.,0.,0.,0.,
  0.,0.,0.,0.,0.,
  12586688.,1320.0366,0.,145326.19,
  236.97725,-33.303009,33.303009,
  3666.41558,0.,
  3666.41558,
  XII=-202.36484,6560.4276,-.0003921024,-9.0054416,-.27778424,-4.8975852
 &END
/*
```

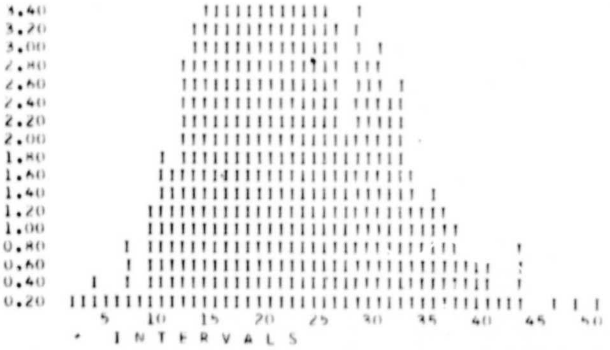
Figure 1. Sample Input and J.C.L. Cards



```
//G7DPMRA9 JOB (G70031150A,T,D00130,001001),EFF,MSGLEVEL=1
// EXEC LOADER,PARM='SIZE=400K,NOMAP',REGION=400K
XXGO       EXEC PGM=LOADER,PARM=(MAP,CALL),COND=(4,LT),REGION=240K          00000100
XXSYSLIB DD DSN=SYS2.DUMMY,DISP=SHR                                           00000200
XX          DD DSN=SYS2.DUMMY,DISP=SHR                                        00000300
XX          DD DSN=SYS1.FORTLIB,DISP=SHR                                      00000400
XX          DD DSN=SYS2.GSFCLIB,DISP=SHR                                      00000500
XX          DD DSN=SYS1.PL1LIB,DISP=SHR                                       00000600
XX          DD DSN=SYS1.TELCMLIB,DISP=SHR                                     00000700
XX          DD DSN=SYS2.LOADLIB,DISP=SHR                                      00000800
XX          DD DSN=SYS1.SSPAK,DISP=SHR                                        00000900
XXSYSLOUT  DD  SYSOUT=A                                                       00001000
//GO.SYSLIN DD UNIT=2400-9,LABEL=(3,BLP),VOL=SER=30704G,
//    DISP=(OLD,PASS),DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DEN=3)
X/SYSLIN    DD  DSN=&&OBJMOD,DISP=(OLD,DELETE)                                00001100
XX          DD  DDNAME=OBJECT                                                 00001200
XXFT05F001 DD  DDNAME=DATA5                                                   00001300
XXFT06F001 DD  SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),               00001400
XX              SPACE=(CYL,(1,1))                                             00001500
XXFT07F001 DD  DUMMY,DCB=(RECFM=FB,LRECL=80,BLKSIZE=[illegible]280),          00001600
XX              SPACE=(TRK,(1,20))                                            00001700
***   INSERT  //GO.FT07F001 DD DSN=&&DECK,SYSOUT=B  FOR PUNCHED OUTPUT  00001750
XXSYSPRINT DD  SYSOUT=A,DCB=(RECFM=VBA,LRECL=125,BLKSIZE=629),              0000180
XX              SPACE=(TRK,(1,20))                                            00001900
//GO.DATA5              DD UNIT=2314,DISP=(OLD,DELETE),VOLUME=SER=G1SCR1, X
//              DSNAME=SYS71137.T123456.F0007.G7DPMRA9.S0000019,        X
//              SPACE=(80,(000013))
//
IEF236I ALLOC. FOR G7DPMRA9 GO
IEF237I 547   ALLOCATED TO SYSLIB
IEF237I 547   ALLOCATED TO
IEF237I 547   ALLOCATED TO
IEF237I 547   ALLOCATED TO
IEF237I 547   ALLOCATED TO
IEF237I 547   ALLOCATED TO
IEF237I 547   ALLOCATED TO
IEF237I 337   ALLOCATED TO
IEF237I 331   ALLOCATED TO SYSLOUT
IEF237I 0C3   ALLOCATED TO SYSLIN
IEF237I 230   ALLOCATED TO FT05F001
IEF237I 332   ALLOCATED TO FT06F001
IEF237I 333   ALLOCATED TO SYSPRINT
                                                                   OS/360 LOADER

OPTIONS USED - PRINT,NOMAP,NOLET,CALL,RES,SIZE=401408

  TOTAL LENGTH     55088
  ENTRY ADDRESS  245008



           SAS-D ERROR PROPAGATION

                      DTC                           DTB                             W                            WC                              PITCH
 0.19126184000000000 05   0.47960240000000000D 02   0.12250000000000000 04  -0.60154990000000000D 03   0.0

                      YAW                        THRUST                         ZDELT                         NCONF                              IHIST
-0.51314170000000000D 02   0.34530000000000000D 04        0.0                                                1000                      1 2 3 7 8 9

                       AS                            FS                           AIS                         NBURN                               XMU
 0.42164000000000000D 05   0.0                           -0.10000000000000000D 01                             50   0.39860320000000000D 06

    SAMPLE INTERVAL MATRIX
     0.0                  0.0                  0.0                  0.0                  0.0                  0.0
     0.50000000E 02       0.50000000F 02       0.50000000F 02       0.50000000E 02       0.50000000F 02       0.50000000F 02
     0.0                  0.0                  0.0                  0.0                  0.0                  0.0

    SIGB
 0.86325000000000000D 01   0.16670000000000000D 00   0.16670000000000000D 00

    INITIAL STATE VECTOR
-0.20236484000000D 03  0.65604276000000D 04 -0.39210240000000D-03 -0.90054416000000D 01 -0.27778424000000D 00 -0.48975852000000D 01

    INITIAL COVARIANCE MATRIX      IPCOOR = 1
 0.0                   0.0                   0.0                   0.0                   0.0                   0.0
 0.0                   0.0                   0.0                   0.0                   0.0                   0.0
 0.0                   0.0                   0.12586688000000D 08  0.13200366000000D 04  0.0                   0.14532619000000D 06
 0.0                   0.0                   0.13200366000000D 04  0.23697725000000D 03 -0.33303009000000D 02  0.33303009000000D 02
 0.0                   0.0                   0.0                  -0.33303009000000D 02  0.36664155800000D 04  0.0
 0.0                   0.0                   0.14532619000000D 06  0.33303009000000D 02  0.0                   0.36664155800000D 04


                    BEFORE COAST



    STATE VECTOR
-0.20236484000000D 03  0.65604276000000D 04 -0.39210240000000D-03 -0.90054416000000D 01 -0.27778424000000D 00 -0.48975852000000D 01

    COVARIANCE MATRIX (LOCAL TANGENT)
-0.35822525975580D-20 -0.19472361252030D-20 -0.27000778232800D-17  0.48222628184360D-21  0.56454775385640D-21 -0.12766366933590D-18
-0.18914192863190D-19 -0.10281568712620D-19  0.16502468550940D-17  0.35113813467530D-21  0.15253495035990D-21 -0.25886597526710D-19
 0.56200793214640D-17  0.62796772778170D-17  0.11693411753690D 01  0.12263537074840D-03 -0.71150767569360D-19  0.13501240185390D-01
-0.20696301264150D-20 -0.92717495482460D-21  0.12263537074840D-03  0.22015899341500D-04 -0.30939497099950D-05  0.30939497099950D-05
 0.43246624240350D-21  0.23508170484220D-21 -0.91507885586940D-19 -0.30939497099950D-05  0.34062103578880D-03  0.14743654313450D-20
-0.10338578924540D-18 -0.12778906488180D-20  0.13501240185390D-01  0.30939497099950D-05  0.29646153153900D-20  0.34062103578880D-03

    COVARIANCE MATRIX (LOC TAN -- FT)
-0.38559059351010D-13 -0.20959882441940D-13 -0.29063405833290D-10  0.51906422888490D-14  0.60767435446250D-14 -0.13741607742140D-11
-0.20359074781190D-12 -0.11066992274130D-12  0.17763115441040D-10  0.37796207293080D-14  0.16418731074520D-14 -0.27864111288760D-12
```

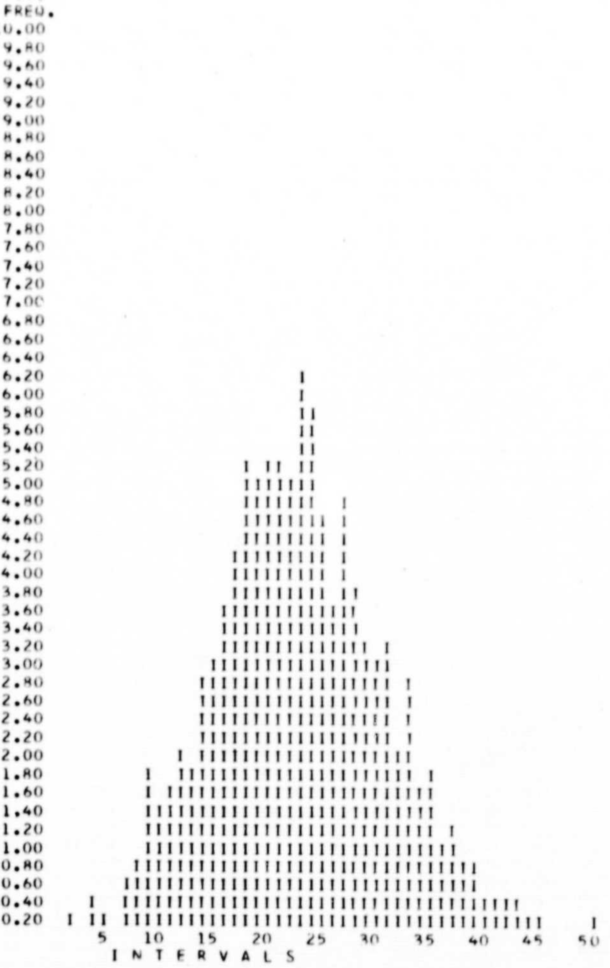
Figure 2. Sample Printer Output

```
  0.6049405121834D-10  0.6759390698065D-10  0.1258668800000D 08  0.1320036600000D 04 -0.7658607523789D-12  0.1453261900000D 06
 -0.2227732096097D-13 -0.9980031596941D-14  0.1320036600000D 04  0.2369772500000D 03 -0.3330300900000D 02  0.3330300900000D 02
  0.4655029497225D-14  0.2530399284385D-14 -0.9849830226482D-12 -0.3330300900000D 02  0.3666415580000D 04  0.1586994289879D-13
 -0.1112835757669D-11 -0.1375510521102D-13  0.1453261900000D 06  0.3330300900000D 02  0.3191086468245D-13  0.3666415580000D 04

    COVARIANCE MATRIX (INERTIAL)
  0.1111562872689D-02 -0.3603554722808D-01  0.2153765793169D-08  0.1615451833725D-04 -0.4159648378242D-03  0.1805808829861D-05
 -0.3603554722808D-01  0.1168229612497D 01 -0.6982252724061D-07 -0.5237102846739D-03  0.1348508566355D-01 -0.5854217604077D-04
  0.2153765793169D-08 -0.6982252724061D-07  0.4173139644908D-14  0.3130101756253D-10 -0.8059740576796D-09  0.3498937741011D-11
  0.1615451833725D-04 -0.5237102846739D-03  0.3130101756253D-10  0.9249406044156D-04 -0.1037209806716D-04 -0.1318962674725D-03
 -0.4159648378242D-03  0.1348508566355D-01 -0.8059740576796D-09 -0.1037209806716D-04  0.3402172453632D-03 -0.5546851348534D-05
  0.1805808829861D-05 -0.5854217604077D-04  0.3498937741011D-11 -0.1318962674725D-03 -0.5546851348534D-05  0.2705466651143D-03
    EIGENVALUES
  0.1169497098494D 01  0.3406510785498D-03  0.1847280672124D-03  0.2195570095872D-04  0.5193610709948D-15  0.0

    ORBITAL ELEMENTS NOMINAL
          ECC                  SMA                  INC                  OMEGA                ARGP                 THETA
  0.7316279425124D 00  0.2445689771674D 05  0.2852800600174D 02  0.2717667955874D 03  0.1800000005112D 03  0.6537085616600D-05

    TRANSITION MATRIX
  0.6393969923715D 01 -0.4350860501579D 02  0.6250518845275D 01  0.4275434540038D 05 -0.7061413555165D 04  0.2311932731741D 05
  0.1923714158615D 01 -0.5656068131344D 02  0.2054107681845D 00  0.5606346615712D 05  0.1439322362276D 04  0.3048534954120D 05
  0.7011949267136D 01 -0.2447934657000D 02 -0.3052650956483D 01  0.2417727911610D 05 -0.3812312696075D 04  0.1305774019147D 05
  0.3119343624004D-04 -0.1338029770022D-02 -0.7211664716354D-05  0.1386328910817D 01  0.5721307845877D-01  0.8383981661966D 00
  0.4705476333235D-03 -0.7291580677501D-02  0.3334048392420D-03  0.7121405716733D 01 -0.1141784154495D 00  0.3869959578331D 01
  0.2309948558861D-04 -0.8492501328173D-03  0.1510911436248D-05  0.9567916640065D 00  0.3176940369816D-01  0.3654074824474D 00


                        AFTER COAST


    STATE VECTOR
  0.1437319624770D 04 -0.4232506585976D 05  0.7158545856889D 02  0.1395007203445D 01  0.6400081365747D-01  0.7590223676722D 00

    COVARIANCE MATRIX (LOCAL TANGENT)
  0.1244701016238D 06 -0.2601010773965D 01 -0.9167900224503D 05  0.3146169186126D 01 -0.2757955557998D-01 -0.1327003113849D 02
 -0.2601010773990D 01  0.7276564622586D-01  0.2808491110043D 01 -0.9745014127896D-04  0.7715631955926D-03  0.3634676930181D-03
 -0.9167900224503D 05  0.2808491110020D 01  0.8929428049732D 05 -0.3087571089683D 01  0.2977955241085D-01  0.1187787707928D 02
  0.3146169186126D 01 -0.9745014127871D-04 -0.3087571089683D 01  0.1067859472742D-03 -0.1033302750820D-05 -0.4098555160745D-03
 -0.2757955557998D-01  0.7715631955927D-03  0.2977955241085D-01 -0.1033302750820D-05  0.8181192576789D-05  0.3853993048134D-05
 -0.1327003113849D 02  0.3634676930164D-03  0.1187787707928D 02 -0.4098555160745D-03  0.3853993048133D-05  0.1618076112054D-02

    COVARIANCE MATRIX (LOC TAN -- FT)
  0.1339785485593D 13 -0.2799705662138D 08 -0.9868249076621D 12  0.3386509495697D 08 -0.2968639679982D 06 -0.1428374756731D 09
 -0.2799705662165D 08  0.7832431675673D 06  0.3023035714253D 08 -0.1048944952654D 04  0.8305039982996D 04  0.3912335036567D 04
 -0.9868249076621D 12  0.3023035714229D 08  0.9611559675465D 12 -0.3323435007869D 08  0.3205445449713D 06  0.1278524489245D 09
  0.3386509495697D 08 -0.1048944952651D 04 -0.3323435007869D 08  0.1149434766718D 04 -0.1112238207981D 02 -0.4411664958061D 04
 -0.2968639679982D 06  0.8305039982997D 04  0.3205445449713D 06 -0.1112238207981D 02  0.8806165437510D 02  0.4148405022657D 02
 -0.1428374756731D 09  0.3912335036549D 04  0.1278524489245D 09 -0.4411664958061D 04  0.4148405022657D 02  0.1741683232547D 06

    COVARIANCE MATRIX (INERTIAL)
  0.9060404649475D 05  0.8067676347196D 05  0.5057751221668D 05  0.1963119899430D 01  0.1132406090411D 02  0.1229578375293D 01
  0.8067676347196D 05  0.9492019521589D 05  0.4542167713995D 05  0.2392011869776D 01  0.1236823966213D 02  0.1472090494547D 01
  0.5057751221668D 05  0.4542167713995D 05  0.2824021317609D 05  0.1106209350732D 01  0.6359563599291D 01  0.6934582670413D 00
  0.1963119899430D 01  0.2392011869776D 01  0.1106209350732D 01  0.6237081280002D-04  0.3091497787046D-03  0.3374031320986D-04
  0.1132406090411D 02  0.1236823966213D 02  0.6359563599291D 01  0.3091497787046D-03  0.1664151355401D-02  0.1909244735887D-03
  0.1229578375293D 01  0.1472090494547D 01  0.6934582670413D 00  0.3374031320986D-04  0.1909244735887D-03  0.2915888509524D-04
    EIGENVALUES
  0.2002330045402D 06  0.1353137939669D 05  0.7268285434336D-01  0.1256650216134D-07  0.8912412880384D-10  0.0

    ORBITAL ELEMENTS NOMINAL
          ECC                  SMA                  INC                  OMEGA                ARGP                 THETA
  0.7316279425124D 00  0.2445689771674D 05  0.2852800600174D 02  0.2717667955874D 03  0.1800000001497D 03 -0.1797972107338D 03


                        AFTER BURN


    STATE VECTOR
  0.1539992935689D 04 -0.4232020327820D 05  0.9294326896789D 02  0.3074136864709D 01  0.1492290834618D 00  0.5271056778243D-01

    COVARIANCE MATRIX (LOCAL TANGENT)
  0.9849053131864D 05  0.5129859054681D 05 -0.8194265517668D 05  0.2499805647150D 01  0.1265931905956D 01 -0.1184706436972D 02
  0.5129859054681D 05  0.2671899235181D 05 -0.4266059413649D 05  0.1300808384882D 01  0.6602158421203D 00 -0.6168684250144D 01
 -0.8194265517668D 05 -0.4266059413649D 05  0.9000769964050D 05 -0.2770757336168D 01 -0.1401756311074D 01  0.1196188506067D 02
  0.2499805647150D 01  0.1300808384882D 01 -0.2770757336168D 01  0.8707635232960D-04  0.3981585367803D-04 -0.3673232780725D-03
  0.1265931905956D 01  0.6602158421203D 00 -0.1401756311074D 01  0.3981585367803D-04  0.2825992673078D-04 -0.1859147906335D-03
 -0.1184706436972D 02 -0.6168684250144D 01  0.1196188506067D 02 -0.3673232780725D-03 -0.1859147906335D-03  0.1628123079046D-02

    COVARIANCE MATRIX (LOC TAN -- FT)
  0.1060143621702D 13  0.5521736236205D 12 -0.8820237038815D 12  0.2690769332687D 08  0.1362638232975D 08 -0.1275207835647D 09
  0.5521736236205D 12  0.2876009393070D 12 -0.4591949720071D 12  0.1400178975406D 08  0.7106506631651D 07 -0.6639918756179D 08
 -0.8820237038815D 12 -0.4591949720071D 12  0.9687605594999D 12 -0.2982419404075D 08 -0.1508838456317D 08  0.1287567036224D 09
  0.2690769332687D 08  0.1400178975406D 08 -0.2982419404075D 08  0.9372823792033D 03  0.4285744299910D 03 -0.3953836223012D 04
  0.1362638232975D 08  0.7106506631651D 07 -0.1508838456317D 08  0.4285744299910D 03  0.3041874246418D 03 -0.2001170841820D 04
 -0.1275207835647D 09 -0.6639918756179D 08  0.1287567036224D 09 -0.3953836223012D 04 -0.2001170841820D 04  0.1752497701546D 05

    COVARIANCE MATRIX (INERTIAL)
  0.9079130084907D 05  0.8133566931693D 05  0.5068897322679D 05  0.1941204477512D 01  0.1137707060189D 02  0.1218319513673D 01
  0.8133566931693D 05  0.9611254016242D 05  0.4579796164958D 05  0.2383957425051D 01  0.1249303644178D 02  0.1469605346392D 01
  0.5068897322679D 05  0.4579796164958D 05  0.2830645262301D 05  0.1093912574031D 01  0.6390235998488D 01  0.6876656306563D 00
  0.1941204477512D 01  0.2383957425051D 01  0.1093912574031D 01  0.6126576497720D-04  0.3071222742943D-03  0.3311420023232D-04
  0.1137707060189D 02  0.1249303644178D 02  0.6390235998488D 01  0.3071222742943D-03  0.1653386651754D-02  0.1900074238466D-03
  0.1218319513673D 01  0.1469605346392D 01  0.6876656306563D 00  0.3311420023232D-04  0.1900074238466D-03  0.2880694137713D-04
    EIGENVALUES
  0.2016499197350D 06  0.1356021025067D 05  0.1653922378077D 00  0.1256539225451D-07  0.8975280181794D-10  0.0

                        BURN ERRORS

    STATE VECTOR
  0.1539992935689D 04 -0.4232020327820D 05  0.9294326896789D 02  0.3074136864709D 01  0.1492290834618D 00  0.5271056778243D-01

    COVARIANCE MATRIX (LOCAL TANGENT)
  0.1006910450438D-01  0.1189855134227D-02 -0.2523672138194D-04  0.4730540617993D-03  0.5570549348903D-04 -0.1450793444308D-05
 0.11898 .13^ 27D-02   0.1224711694383D-01 -0.2416291614625D-04  0.5570912099915D-04  0.5750235669712D-03 -0.6884875439953D-06
-0.2523672138194D-04 -0.2416291614625D-04  0.7411183527649D-02 -0.5681258710152D-06 -0.1366478592163D-05  0.3716309712838D-03
 0.4730540617993D-03  0.5570912099915D-04 -0.5681258710152D-06  0.2222448810691D-04  0.2609090796020D-05 -0.3720291403300D-07
 0.5570549348903D-04  0.5750235669712D-03 -0.1366478592163D-05  0.2608090796020D-05  0.2699837223939D-04 -0.4386994899880D-07
-0.1450793444308D-05 -0.6884875439953D-06  0.3716309712838D-03 -0.3720291403300D-07 -0.4386994899880D-07  0.1863532125993D-04

    COVARIANCE MATRIX (LOC TAN -- FT)
 0.1083829762481D 06  0.1280749849161D 05 -0.2716659018704D 03  0.5091913299914D 04  0.5996091484685D 03 -0.1561621605446D 02
 0.1280749849161D 05  0.1318269151197D 06 -0.2600875545214D 03  0.5996481946760D 03  0.6189504297432D 04 -0.7410820802933D 01
-0.2716659018704D 03 -0.2600875545214D 03  0.7977334309167D 05 -0.6115258090469D 01 -0.1469043846556D 02  0.4000203862836D 04
 0.5091913299914D 04  0.5996481946760D 03 -0.6115258090469D 01  0.2392224815593D 03  0.2807326537082D 02 -0.4004489720259D 00
 0.5996091484685D 03  0.6189504297432D 04 -0.1469043846556D 02  0.2807326537082D 02  0.2906081604265D 03 -0.4722123638974D 00
-0.1561621605446D 02 -0.7410820802933D 01  0.4000203862836D 04 -0.4004489720259D 00 -0.4722123638974D 00  0.2005889978212D 03

    COVARIANCE MATRIX (INERTIAL)
 0.1002348618284D-01  0.1224009067097D-03  0.1149701638147D-02  0.4709610894651D-03  0.5171444921201D-05  0.5381872057423D-04
 0.1224009067097D-03  0.7416764127848D-02  0.7715363125203D-04  0.4277317691280D-05  0.3718488514259D-03  0.3784459513650D-05
 0.1149701638147D-02  0.7715363125203D-04  0.1228715466516D-01  0.5384834621053D-04  0.3124040315176D-05  0.5768986591634D-03
 0.4709610894651D-03  0.4277317691280D-05  0.5384834621053D-04  0.2212875865099D-04  0.1691002806382D-06  0.2520639833198D-05
 0.5171444921201D-05  0.3718488514259D-03  0.3124040315176D-05  0.1691002806382D-06  0.1866323853686D-04  0.1548144988799D-06
 0.5381872057423D-04  0.3784459513650D-05  0.5768986591634D-03  0.2520639833198D-05  0.1548144988799D-06  0.27086;8461837D-04
    EIGENVALUES
 0.1279942785684D-01  0.9566294941704D-02  0.7429540358914D-02  0.7825772039796D-17  0.7825772039796D-17  0.0


                         AFTER BURN WITH BURN ERRORS



    STATE VECTOR
 0.1539992935689D 04 -0.4232020327820D 05  0.9294326896789D 02  0.3074136864709D 01  0.1492290834618D 00  0.5271056778243D-01

    COVARIANCE MATRIX (LOCAL TANGENT)
 0.9849054138774D 05  0.5129859173667D 05 -0.8194265520192D 05  0.2500278701212D 01  0.1265987611449D 01 -0.1184706582051D 02
 0.5129859173667D 05  0.2671900459893D 05 -0.4266059416065D 05  0.1300864094003D 01  0.6607908656873D 00 -0.6168684938632D 01
-0.8194265520192D 05 -0.4266059416065D 05  0.9000077737523D 05 -0.2770757904294D 01 -0.1401757675860D 01  0.1196225669164D 02
 0.2500278701212D 01  0.1300864094003D 01 -0.2770757904294D 01  0.1093008404365D-03  0.4242394447405D-04 -0.3673604809865D-03
 0.1265987611449D 01  0.6607908656873D 00 -0.1401757675860D 01  0.4242394447405D-04  0.5525829897017D-04 -0.1859586605825D-03
-0.1184706582051D 02 -0.6168684938632D 01  0.1196225669164D 02 -0.3673604809865D-03 -0.1859586605825D-03  0.1646758400308D-02

    COVARIANCE MATRIX (LOC TAN -- FT)
 0.1060143730085D 13  0.5521736364280D 12 -0.8820237041531D 12  0.2691278524017D 08  0.1362698193890D 08 -0.1275207991809D 09
 0.5521736364280D 12  0.2876010711339D 12 -0.4591949722672D 12  0.1400238940226D 08  0.7112696135948D 07 -0.6639919497261D 08
-0.8820237041531D 12 -0.4591949722672D 12  0.9687606392733D 12 -0.2982420015600D 08 -0.1508839925361D 08  0.1287607038263D 09
 0.2691278524017D 08  0.1400238940226D 08 -0.2982420015600D 08  0.1176504860763D 04  0.4566647695361D 03 -0.3954236671984D 04
 0.1362698193890D 08  0.7112696135948D 07 -0.1508839925361D 08  0.4566647695361D 03  0.5947955850683D 03 -0.2001643054184D 04
-0.1275207991809D 09 -0.6639919497261D 08  0.1287607038263D 09 -0.3954236671984D 04 -0.2001643054184D 04  0.1772556601328D 05

    COVARIANCE MATRIX (INERTIAL)
 0.9079131087256D 05  0.8133566943933D 05  0.5068897437649D 05  0.1941675438601D 01  0.1137707577333D 02  0.1218373332394D 01
 0.8133566943933D 05  0.9611254757918D 05  0.4579796172673D 05  0.2383961702369D 01  0.1249340829063D 02  0.1469609130852D 01
 0.5068897437649D 05  0.4579796172673D 05  0.2830666491016D 05  0.1093966422378D 01  0.6390239122528D 01  0.6882425293155D 00
 0.1941675438601D 01  0.2383961702369D 01  0.1093966422378D 01  0.8339452342819D-04  0.3072913745750D-03  0.3563484006552D-04
 0.1137707577333D 02  0.1249340829063D 02  0.6390239122528D 01  0.3072913745750D-03  0.1672029890291D-02  0.1901622383453D-03
 0.1218373332394D 01  0.1469609130852D 01  0.6882425293155D 00  0.3563484006552D-04  0.1901622383453D-03  0.5589312599550D-04
    EIGENVALUES
 0.2016499296115D 06  0.1356021929615D 05  0.1762055032952D 00  0.2612282357965D-04  0.1856940689329D-04  0.1541764734151D-04

    ORBITAL ELEMENTS NOMINAL
          ECC                  SMA                  INC                  OMEGA                ARGP                 THETA
 0.1381461198375D-01  0.4263309217392D 05  0.9907751783161D 00  0.2647930994767D 03  0.6907130564491D 02 -0.6177930634377D 02

          PER                  APOG                 DELV
 0.4204413254787D 05  0.4322205179997D 05  0.2116643656072D-01


          SAS-D ERROR PROPAGATION
    ECC        NOMINAL=  1.3815E-02  MEAN=  1.4731E-02  SIGMA=  5.0621E-03  SAMPLE=  1000

     PCT.                                                             INTERVALS            PCT. FREQ.     SUM
     FREQ.                                                  1   MINUS INF.   1.2164E-03   0.0          0.0
    10.00                                                   2   1.2164E-03   1.8785E-03   0.2000000    0.2000000
     9.80                                                   3   1.8785E-03   2.5407E-03   0.1000000    0.3000000
     9.60                                                   4   2.5407E-03   3.2029E-03   0.3000000    0.5999999
     9.40                                                   5   3.2029E-03   3.8650E-03   0.2000000    0.7999999
     9.20                                                   6   3.8650E-03   4.5272E-03   0.2000000    0.9999999
     9.00                                                   7   4.5272E-03   5.1893E-03   0.8000000    1.7999992
     8.80                                                   8   5.1893E-03   5.8515E-03   0.2000000    1.9999990
     8.60                                                   9   5.8515E-03   6.5137E-03   1.1999998    3.1999989
     8.40                                                  10   6.5137E-03   7.1758E-03   1.7999992    4.9999981
     8.20                                                  11   7.1758E-03   7.8380E-03   1.5999994    6.5999975
     8.00                                                  12   7.8380E-03   8.5002E-03   2.6999998    9.2999973
     7.80                                                  13   8.5002E-03   9.1623E-03   3.1999998   12.4999971
     7.60                                                  14   9.1623E-03   9.8245E-03   3.8999996   16.3999939
     7.40                                                  15   9.8245E-03   1.0487E-02   4.1999998   20.5999908
     7.20                                                  16   1.0487E-02   1.1149E-02   4.5999994   25.1999817
     7.00                                                  17   1.1149E-02   1.1811E-02   5.7999992   30.9999695
     6.80                                                  18   1.1811E-02   1.2473E-02   5.0999994   36.0999603
     6.60                                                  19   1.2473E-02   1.3135E-02   4.6999998   40.7999573
     6.40                                                  20   1.3135E-02   1.3797E-02   5.5000000   46.2999573
     6.20                                                  21   1.3797E-02   1.4460E-02   5.0000000   51.2999573
     6.00                                                  22   1.4460E-02   1.5122E-02   4.5000000   55.7999573
     5.80                      I                           23   1.5122E-02   1.5784E-02   5.1999998   60.9999542
     5.60                      I  I                        24   1.5784E-02   1.6446E-02   5.3999996   66.3999481
     5.40                      I  I   I                    25   1.6446E-02   1.7108E-02   4.6999998   71.0999451
     5.20                      II I  II                    26   1.7108E-02   1.7770E-02   3.1999998   74.2999420
     5.00                      II II II                    27   1.7770E-02   1.8433E-02   2.0000000   76.2999420
     4.80                      IIIII III                   28   1.8433E-02   1.9095E-02   4.0000000   80.2999420
     4.60                     IIIIIIIIII                   29   1.9095E-02   1.9757E-02   2.7999992   83.0999298
     4.40                     IIIIIIIIII                   30   1.9757E-02   2.0419E-02   2.8999996   85.9999237
     4.20                    IIIIIIIIIII                   31   2.0419E-02   2.1081E-02   2.2999992   88.2999115
     4.00                   IIIIIIIIIIII  I                32   2.1081E-02   2.1743E-02   2.5999994   90.8999023
     3.80                   IIIIIIIIIIII  I                33   2.1743E-02   2.2406E-02   1.5999994   92.4998932
     3.60                   IIIIIIIIIIII  I                34   2.2406E-02   2.3068E-02   1.1999998   93.6998901
```



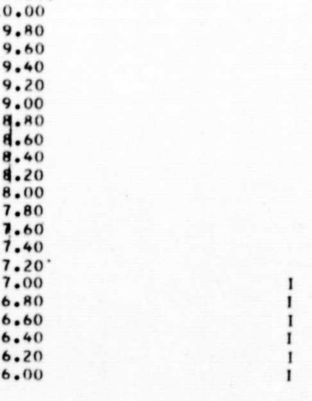
```
35   2.3068E-02  2.3730E-02    1.3999996  95.0998840
36   2.3730E-02  2.4392E-02    1.0999994  96.1998749
37   2.4392E-02  2.5054E-02    0.9000000  97.0998688
38   2.5054E-02  2.5716E-02    0.6000000  97.6998596
39   2.5716E-02  2.6379E-02    0.5000000  98.1998596
40   2.6379E-02  2.7041E-02    0.5000000  98.6998596
41   2.7041E-02  2.7703E-02    0.2000000  98.8998566
42   2.7703E-02  2.8365E-02    0.1000000  98.9998474
43   2.8365E-02  2.9027E-02    0.7000000  99.6998444
44   2.9027E-02  2.9689E-02    0.0        99.6998444
45   2.9689E-02  3.0352E-02    0.0        99.6998444
46   3.0352E-02  3.1014E-02    0.1000000  99.7998352
47   3.1014E-02  3.1676E-02    0.0        99.7998352
48   3.1676E-02  3.2338E-02    0.1000000  99.8998260
49   3.2338E-02  3.3000E-02    0.0        99.8998260
50   3.3000E-02  PLUS INF.     0.1000000  99.9998169
```

```
        SAS-D ERROR PROPAGATION
SMA       NOMINAL=  4.2633E 04  MEAN=  4.2663E 04  SIGMA=  3.6797E 02  SAMPLE=  1000
```

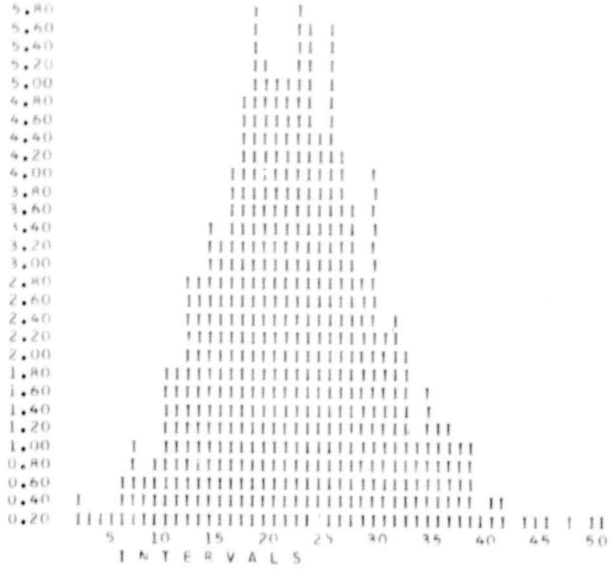


```
           INTERVALS                  PCT. FREQ.     SUM
 1   MINUS INF.     4.1621E 04    0.0            0.0
 2    4.1621E 04    4.1670E 04    0.1000000      0.1000000
 3    4.1670E 04    4.1719E 04    0.0            0.1000000
 4    4.1719E 04    4.1767E 04    0.4000000      0.4999999
 5    4.1767E 04    4.1816E 04    0.1000000      0.5999999
 6    4.1816E 04    4.1865E 04    0.0            0.5999999
 7    4.1865E 04    4.1914E 04    0.6000000      1.1999998
 8    4.1914E 04    4.1963E 04    0.7000000      1.8999996
 9    4.1963E 04    4.2011E 04    1.6999998      3.5999994
10    4.2011E 04    4.2060E 04    1.2999992      4.8999987
11    4.2060E 04    4.2109E 04    1.5999994      6.4999981
12    4.2109E 04    4.2158E 04    2.0000000      8.4999981
13    4.2158E 04    4.2207E 04    1.7999992     10.2999973
14    4.2207E 04    4.2255E 04    2.7999992     13.0999966
15    4.2255E 04    4.2304E 04    2.8999996     15.9999962
16    4.2304E 04    4.2353E 04    3.5999994     19.5999908
17    4.2353E 04    4.2402E 04    4.1999998     23.7999878
18    4.2402E 04    4.2451E 04    5.0999994     28.8999786
19    4.2451E 04    4.2500E 04    5.0000000     33.8999786
20    4.2500E 04    4.2548E 04    5.1999998     39.0999756
21    4.2548E 04    4.2597E 04    5.0999994     44.1999664
22    4.2597E 04    4.2646E 04    5.0000000     49.1999664
23    4.2646E 04    4.2695E 04    6.1999998     55.3999634
24    4.2695E 04    4.2744E 04    5.6999998     61.0999603
25    4.2744E 04    4.2792E 04    4.5999994     65.6999512
26    4.2792E 04    4.2841E 04    3.5000000     69.1999512
27    4.2841E 04    4.2890E 04    4.6999998     73.8999481
28    4.2890E 04    4.2939E 04    3.6999998     77.5999451
29    4.2939E 04    4.2988E 04    3.1999998     80.7999420
30    4.2988E*04    4.3036E 04    2.8999996     83.6999359
31    4.3036E 04    4.3085E 04    3.1999998     86.8999329
32    4.3085E 04    4.3134E 04    2.0000000     88.8999329
33    4.3134E 04    4.3183E 04    2.6999998     91.5999298
34    4.3183E 04    4.3232E 04    1.5999994     93.1999207
35    4.3232E 04    4.3280E 04    1.6999998     94.8999176
36    4.3280E 04    4.3329E 04    0.9000000     95.7999115
37    4.3329E 04    4.3378E 04    1.0999994     96.8999023
38    4.3378E 04    4.3427E 04    0.7000000     97.5998993
39    4.3427E 04    4.3476E 04    0.8000000     98.3998871
40    4.3476E 04    4.3524E 04    0.3000000     98.6998749
41    4.3524E 04    4.3573E 04    0.4000000     99.0998688
42    4.3573E 04    4.3622E 04    0.3000000     99.3998566
43    4.3622E 04    4.3671E 04    0.3000000     99.6998444
44    4.3671E 04    4.3720E 04    0.1000000     99.7998352
45    4.3720E 04    4.3768E 04    0.1000000     99.8998260
46    4.3768E 04    4.3817E 04    0.0           99.8998260
47    4.3817E 04    4.3866E 04    0.0           99.8998260
48    4.3866E 04    4.3915E 04    0.0           99.8998260
49    4.3915E 04    4.3964E 04    0.0           99.8998260
50    4.3964E 04    PLUS INF.     0.1000000     99.9998169
```

```
        SAS-D ERROR PROPAGATION
INC       NOMINAL=  9.9078E-01  MEAN=  1.0107E 00  SIGMA=  1.5342E-01  SAMPLE=  1000
```

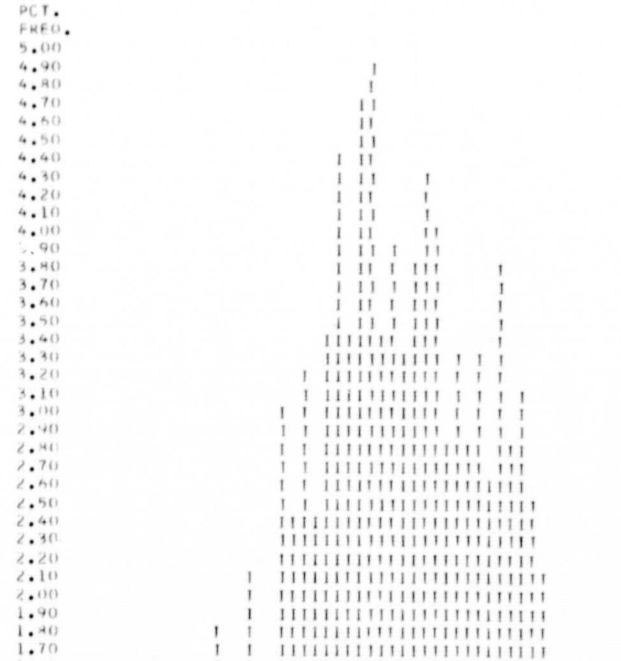


```
           INTERVALS                  PCT. FREQ.     SUM
 1   MINUS INF.     5.9242E-01    0.0            0.0
 2    5.9242E-01    6.1299E-01    0.3000000      0.3000000
 3    6.1299E-01    6.3355E-01    0.2000000      0.4999999
 4    6.3355E-01    6.5412E-01    0.2000000      0.6999999
 5    6.5412E-01    6.7468E-01    0.2000000      0.8999999
 6    6.7468E-01    6.9525E-01    0.6000000      1.4999990
 7    6.9525E-01    7.1582E-01    1.0000000      2.4999990
 8    7.1582E-01    7.3638E-01    0.5000000      2.9999990
 9    7.3638E-01    7.5695E-01    0.8000000      3.7999983
10    7.5695E-01    7.7751E-01    1.7999992      5.5999975
11    7.7751E-01    7.9808E-01    1.6999998      7.2999973
12    7.9808E-01    8.1864E-01    2.6999998      9.9999971
13    8.1864E-01    8.3921E-01    2.7999992     12.7999964
14    8.3921E-01    8.5977E-01    3.3999996     16.1999817
15    8.5977E-01    8.8034E-01    3.0999994     19.2999725
16    8.8034E-01    9.0091E-01    4.0000000     23.2999725
17    9.0091E-01    9.2147E-01    4.6999998     27.9999695
18    9.2147E-01    9.4204E-01    5.7999992     33.7999573
19    9.4204E-01    9.6260E-01    5.1999998     38.9999542
20    9.6260E-01    9.8317E-01    5.0000000     43.9999542
21    9.8317E-01    1.0037E 00    4.8999996     48.8999481
22    1.0037E 00    1.0243E 00    6.8999996     55.7999420
```



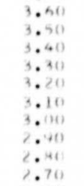
```
23   1.0243F 00   1.0449F 00    5.5000000   61.2999630
24   1.0449F 00   1.0654F 00    4.2999992   65.5999298
25   1.0654F 00   1.0860F 00    5.5000000   71.0999298
26   1.0860F 00   1.1066F 00    4.1999998   75.2999268
27   1.1066F 00   1.1271F 00    3.5999994   78.8999176
28   1.1271F 00   1.1477F 00    2.6999998   81.5999146
29   1.1477F 00   1.1683F 00    3.8999996   85.4999084
30   1.1683F 00   1.1888F 00    2.0999994   87.5998993
31   1.1888F 00   1.2094F 00    2.3999996   89.9998932
32   1.2094F 00   1.2299F 00    1.8999996   91.8998871
33   1.2299F 00   1.2505F 00    1.0000000   92.8998871
34   1.2505F 00   1.2711F 00    1.5000000   94.3998871
35   1.2711F 00   1.2916F 00    1.0999994   95.4998779
36   1.2916F 00   1.3122F 00    1.1999998   96.6998749
37   1.3122F 00   1.3328F 00    0.9000000   97.5998658
38   1.3328F 00   1.3533F 00    0.9000000   98.4998627
39   1.3533F 00   1.3739F 00    0.2000000   98.6998596
40   1.3739F 00   1.3945F 00    0.3000000   98.9998474
41   1.3945F 00   1.4150F 00    0.3000000   99.2998352
42   1.4150F 00   1.4356F 00    0.0          99.2998352
43   1.4356F 00   1.4562F 00    0.1000000   99.3998260
44   1.4562F 00   1.4767F 00    0.2000000   99.5998230
45   1.4767F 00   1.4973F 00    0.1000000   99.6998138
46   1.4973F 00   1.5179F 00    0.0          99.6998138
47   1.5179F 00   1.5384F 00    0.1000000   99.7998047
48   1.5384F 00   1.5590F 00    0.0          99.7998047
49   1.5590F 00   1.5796F 00    0.1000000   99.8997955
50   1.5796F 00   PLUS INF.     0.1000000   99.9997864
```

```
         SAS-D ERROR PROPAGA,ION
PER       NOMINAL=  4.206.E 04   MEAN=  4.2036F 04   SIGMA=  5.2140F 02   SAMPLE=  1000
```

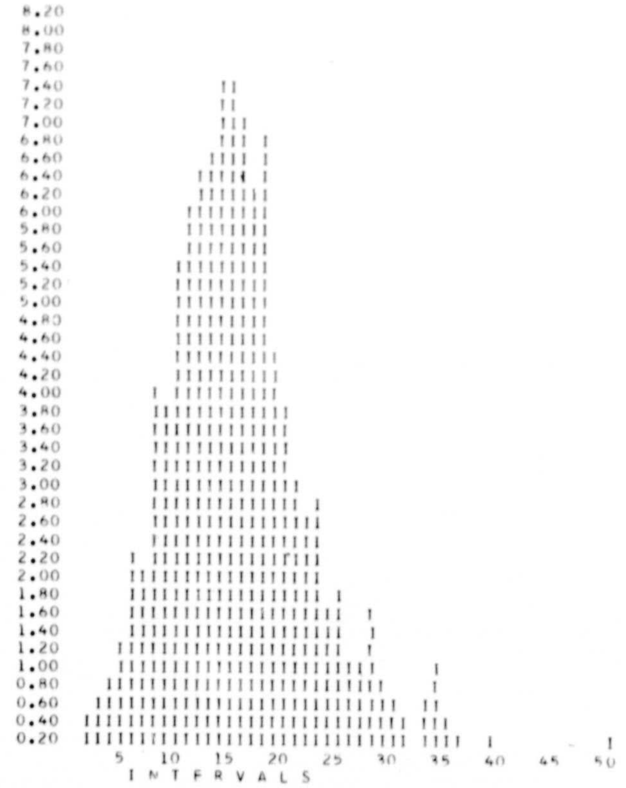


```
            INTERVALS               PCT. FREQ.      SUM
 1   MINUS INF.    4.0432F 04    0.0           0.0
 2   4.0432F 04    4.0489F 04    0.1000000     0.1000000
 3   4.0489F 04    4.0546F 04    0.1000000     0.1999999
 4   4.0546F 04    4.0603F 04    0.3000000     0.4999999
 5   4.0603F 04    4.0660F 04    0.0           0.4999999
 6   4.0660F 04    4.0717F 04    0.0           0.4999999
 7   4.0717F 04    4.0775F 04    0.3000000     0.7999998
 8   4.0775F 04    4.0832F 04    0.6000000     1.3999996
 9   4.0832F 04    4.0889F 04    0.5000000     1.8999996
10   4.0889F 04    4.0946F 04    0.4000000     2.2999992
11   4.0946F 04    4.1003F 04    0.6000000     2.8999987
12   4.1003F 04    4.1060F 04    0.8000000     3.6999979
13   4.1060F 04    4.1117F 04    0.6000000     4.2999973
14   4.1117F 04    4.1174F 04    1.7999992     6.0999966
15   4.1174F 04    4.1231F 04    1.5000000     7.5999966
16   4.1231F 04    4.1288F 04    1.2999992     8.8999958
17   4.1288F 04    4.1345F 04    2.0999994    10.9999952
18   4.1345F 04    4.1402F 04    1.2999992    12.2999945
19   4.1402F 04    4.1460F 04    1.5000000    13.7999945
20   4.1460F 04    4.1517F 04    3.0000000    16.7999878
21   4.1517F 04    4.1574F 04    2.3999996    19.1999817
22   4.1574F 04    4.1631F 04    3.1999998    22.3999786
23   4.1631F 04    4.1688F 04    2.3999996    24.7999725
24   4.1688F 04    4.1745F 04    3.3999996    28.1999664
25   4.1745F 04    4.1802F 04    4.3999996    32.5999603
26   4.1802F 04    4.1859F 04    3.3999996    35.9999542
27   4.1859F 04    4.1916F 04    4.6999998    40.6999512
28   4.1916F 04    4.1973F 04    4.8999996    45.5999451
29   4.1973F 04    4.2030F 04    3.3999996    48.9999390
30   4.2030F 04    4.2087F 04    3.8999996    52.8999329
31   4.2087F 04    4.2144F 04    3.2999992    56.1999207
32   4.2144F 04    4.2202F 04    3.7999992    59.9999084
33   4.2202F 04    4.2259F 04    4.2999992    64.2998962
34   4.2259F 04    4.2316F 04    4.0000000    68.2998962
35   4.2316F 04    4.2373F 04    2.7999992    71.0998840
36   4.2373F 04    4.2430F 04    3.2999992    74.3998718
37   4.2430F 04    4.2487F 04    2.7999992    77.1998596
38   4.2487F 04    4.2544F 04    3.2999992    80.4998474
39   4.2544F 04    4.2601F 04    2.5999994    83.0998383
40   4.2601F 04    4.2658F 04    3.7999992    86.8998260
41   4.2658F 04    4.2715F 04    2.7999992    89.6998138
42   4.2715F 04    4.2772F 04    3.0999994    92.7998047
43   4.2772F 04    4.2829F 04    2.5000000    95.2998047
44   4.2829F 04    4.2887F 04    2.0999994    97.3997955
45   4.2887F 04    4.2944F 04    0.9000000    98.2997894
46   4.2944F 04    4.3001F 04    0.6000000    98.8997803
47   4.3001F 04    4.3058F 04    0.6000000    99.4997711
48   4.3058F 04    4.3115F 04    0.3000000    99.7997589
49   4.3115F 04    4.3172F 04    0.1000000    99.8997498
50   4.3172F 04    PLUS INF.     0.1000000    99.9997406
```

```
         SAS-D ERROR PROPAGATION
APOG      NOMINAL=  4.3222F 04   MEAN=  4.3290F 04   SIGMA=  2.9516F 02   SAMPLE=  1000
```

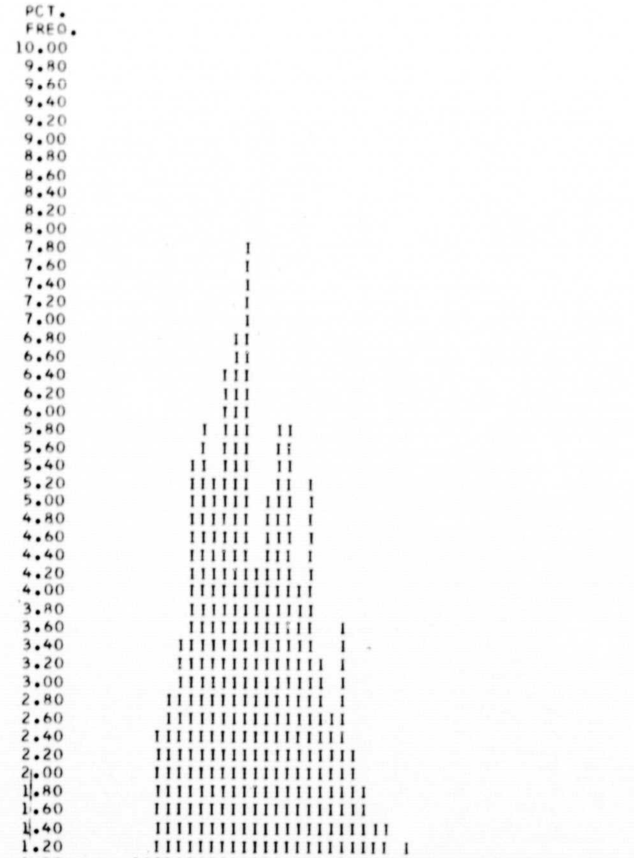


```
            INTERVALS               PCT. FREQ.      SUM
 1   MINUS INF.    4.2623F 04    0.0           0.0
 2   4.2623F 04    4.2670F 04    0.3000000     0.3000000
 3   4.2670F 04    4.2717F 04    0.5000000     0.8000000
 4   4.2717F 04    4.2764F 04    0.7000000     1.4999990
 5   4.2764F 04    4.2811F 04    1.0999994     2.5999985
 6   4.2811F 04    4.2858F 04    2.0999994     4.6999979
 7   4.2858F 04    4.2905F 04    2.0000000     6.6999979
 8   4.2905F 04    4.2951F 04    3.8999996    10.5999975
 9   4.2951F 04    4.2998F 04    3.7999992    14.3999968
10   4.2998F 04    4.3045F 04    5.3999996    19.7999878
```



```
11    4.3045E 04   4.3092E 04    5.8999996   25.6999817
12    4.3092E 04   4.3139E 04    6.2999992   31.9999805
13    4.3139E 04   4.3186E 04    6.5999994   38.5999803
14    4.3186E 04   4.3233E 04    7.3999996   45.9999862
15    4.3233E 04   4.3280E 04    7.2999992   53.2999620
16    4.3280E 04   4.3326E 04    7.0000000   60.2999620
17    4.3326E 04   4.3373E 04    6.0999994   66.3999324
18    4.3373E 04   4.3420E 04    6.6999998   73.0999298
19    4.3420E 04   4.3467E 04    4.2999992   77.3999176
20    4.3467E 04   4.3514E 04    3.6999998   81.0999168
21    4.3514E 04   4.3561E 04    2.8999996   83.9999084
22    4.3561E 04   4.3608E 04    2.5000000   86.4999084
23    4.3608E 04   4.3654E 04    2.6999998   89.1999084
24    4.3654E 04   4.3701E 04    1.5999994   90.7998962
25    4.3701E 04   4.3748E 04    1.7999992   92.5998840
26    4.3748E 04   4.3795E 04    1.0000000   93.5998840
27    4.3795E 04   4.3842E 04    1.0000000   94.5998840
28    4.3842E 04   4.3889E 04    1.5000000   96.0998840
29    4.3889E 04   4.3936E 04    0.7000000   96.7998810
30    4.3936E 04   4.3983E 04    0.6000000   97.3998718
31    4.3983E 04   4.4029E 04    0.4000000   97.7998657
32    4.4029E 04   4.4076E 04    0.0         97.7998657
33    4.4076E 04   4.4123E 04    0.5000000   98.2998657
34    4.4123E 04   4.4170E 04    1.0000000   99.2998657
35    4.4170E 04   4.4217E 04    0.3000000   99.5998535
36    4.4217E 04   4.4264E 04    0.2000000   99.7998505
37    4.4264E 04   4.4311E 04    0.0         99.7998505
38    4.4311E 04   4.4357E 04    0.0         99.7998505
39    4.4357E 04   4.4404E 04    0.1000000   99.8998413
40    4.4404E 04   4.4451E 04    0.0         99.8998413
41    4.4451E 04   4.4498E 04    0.0         99.8998413
42    4.4498E 04   4.4545E 04    0.0         99.8998413
43    4.4545E 04   4.4592E 04    0.0         99.8998413
44    4.4592E 04   4.4639E 04    0.0         99.8998413
45    4.4639E 04   4.4686E 04    0.0         99.8998413
46    4.4686E 04   4.4732E 04    0.0         99.8998413
47    4.4732E 04   4.4779E 04    0.0         99.8998413
48    4.4779E 04   4.4826E 04    0.0         99.8998413
49    4.4826E 04   4.4873E 04    0.0         99.8998413
50    4.4873E 04   PLUS INF.     0.1000000   99.9998322
```

```
            SAS-D ERROR PROPAGATION
DELV        NOMINAL=  2.1166E-02  MEAN=  2.8092E-02  SIGMA=  7.5015E-03  SAMPLE=  1000
```



```
          INTERVALS                  PCT. FREQ.     SUM
 1   MINUS INF.    9.9034E-03     0.0           0.0
 2   9.9034E-03    1.1015E-02     0.1000000     0.1000000
 3   1.1015E-02    1.2126E-02     0.1000000     0.1999999
 4   1.2126E-02    1.3238E-02     0.1000000     0.2999999
 5   1.3238E-02    1.4349E-02     0.6000000     0.8999999
 6   1.4349E-02    1.5460E-02     1.0000000     1.8999996
 7   1.5460E-02    1.6572E-02     1.0000000     2.8999996
 8   1.6572E-02    1.7683E-02     2.2999992     5.1999989
 9   1.7683E-02    1.8795E-02     2.6999998     7.8999987
10   1.8795E-02    1.9906E-02     3.3999996    11.2999983
11   1.9906E-02    2.1017E-02     5.3999996    16.6999969
12   2.1017E-02    2.2129E-02     5.6999998    22.3999939
13   2.2129E-02    2.3240E-02     5.1999998    27.5999908
14   2.3240E-02    2.4351E-02     6.3999996    33.9999847
15   2.4351E-02    2.5463E-02     6.7999992    40.7999725
16   2.5463E-02    2.6574E-02     7.6999998    48.4999605
17   2.6574E-02    2.7686E-02     4.1999998    52.6999664
18   2.7686E-02    2.8797E-02     5.0000000    57.6999664
19   2.8797E-02    2.9908E-02     5.7999992    63.4999542
20   2.9908E-02    3.1020E-02     5.6999998    69.1999512
21   3.1020E-02    3.2131E-02     3.8999996    73.0999451
22   3.2131E-02    3.3243E-02     5.0999994    78.1999359
23   3.3243E-02    3.4354E-02     3.0999994    81.2999268
24   3.4354E-02    3.5465E-02     2.5999994    83.8999176
25   3.5465E-02    3.6577E-02     3.5000000    87.3999176
26   3.6577E-02    3.7688E-02     2.1999998    89.5999146
27   3.7688E-02    3.8800E-02     1.7999992    91.3999023
28   3.8800E-02    3.9911E-02     1.2999992    92.6998901
29   3.9911E-02    4.1022E-02     1.3999996    94.0998840
30   4.1022E-02    4.2134E-02     0.7000000    94.7998810
31   4.2134E-02    4.3245E-02     1.0999994    95.8998718
32   4.3245E-02    4.4356E-02     1.0000000    96.8998718
33   4.4356E-02    4.5468E-02     0.9000000    97.7998657
34   4.5468E-02    4.6579E-02     0.3000000    98.0998535
35   4.6579E-02    4.7691E-02     0.4000000    98.4998474
36   4.7691E-02    4.8802E-02     0.1000000    98.5998383
37   4.8802E-02    4.9913E-02     0.4000000    98.9998322
38   4.9913E-02    5.1025E-02     0.3000000    99.2998199
39   5.1025E-02    5.2136E-02     0.1000000    99.3998108
40   5.2136E-02    5.3248E-02     0.3000000    99.6997986
41   5.3248E-02    5.4359E-02     0.1000000    99.7997894
42   5.4359E-02    5.5470E-02     0.0          99.7997894
43   5.5470E-02    5.6582E-02     0.1000000    99.8997803
44   5.6582E-02    5.7693E-02     0.0          99.8997803
45   5.7693E-02    5.8805E-02     0.0          99.8997803
46   5.8805E-02    5.9916E-02     0.0          99.8997803
47   5.9916E-02    6.1027E-02     0.0          99.8997803
48   6.1027E-02    6.2139E-02     0.0          99.8997803
49   6.2139E-02    6.3250E-02     0.0          99.8997803
50   6.3250E-02    PLUS INF.      0.1000000    99.9997711
```

```
SUMMARY OF ERRORS FOR THIS JOB   ERROR NUMBER   NUMBER OF ERRORS

                                  210           511 OR OVER

IEF285I   SYS2.DUMMY                                  KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS2.DUMMY                                  KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS1.FORTLIB                                KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS2.GSFCLIB                                KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS1.PL1LIB                                 KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS1.TELCMLIB                               KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS2.LOADLIB                                KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS1.SSPAK                                  KEPT
IEF285I   VOL SER NOS= G1SYS2.
IEF285I   SYS71137.T091913.SV000.G70PRRAY.R0000013    SYSOUT
IEF285I   VOL SER NOS= G1SCR8.
IEF285I   SYS71137.T091913.RV000.G70PRRAY.OBJMOD      PASSED
IEF285I   VOL SER NOS= 30704G.
IEF285I   SYS71137.T123456.I0007.G70PRRAY.S0000019    SYSIN
IEF285I   VOL SER NOS= G1SCR1.
IEF285I   SYS71137.T123456.I0007.G70PRRAY.S0000019    DELETED
IEF285I   VOL SER NOS= G1SCR1.
IEF285I   SYS71137.T091913.SV000.G70PRRAY.R0000014    SYSOUT
IEF285I   VOL SER NOS= G1SCR9.
IEF285I   SYS71137.T091913.SV000.G70PRRAY.R0000015    DELETED
IEF285I   VOL SER NOS= G1SCR7.
-----JOB NBR-  114 STEP NBR- 01  G70PRRAY  GO         PGM=LOADER   CARDS=00013   INITIATION TIME=09.21.06.21 DATE=05-17-71
----- CPU=000.2 I/O=000.1 CORE=000.4  CHARGE=000.22          STEP=01  GO         TERMINATION TIME=09.29.38.26 DATE=05-17-71
----- I/O TIME BY DEVICE.  DISK=*****2.00,DRUM=******.00,TAPE=******.14,CELL=******.00,OTHR=******.00
----- STEP REGION SIZE=0400K  MAXIMUM REGION SIZE USED=0400K   PERCENT OF REGION USED=99
IEF285I   SYS71137.T091913.RV000.G70PRRAY.OBJMOD      KEPT
IEF285I   VOL SER NOS= 30704G.


----- CPU=000.2 I/O=000.1 CORE=000.4  CHARGE=000.22    JOB NBR-114  G70PRRAY  360/95      SYSTEM=MVT-19.6   01-16-71  G1
----- I/O TIME BY DEVICE.  DISK=*****2.00,DRUM=******.00,TAPE=******.14,CELL=******.00,OTHR=******.00
```

```
SUMMARY OF ERRORS FOR THIS JOB   ERROR NUMBER   NUMBER OF ERRORS

                                     210           511 OR OVER

IEF285I   SYS2.DUMMY                                     KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS2.DUMMY                                     KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS1.FORTLIB                                   KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS2.GSPCLIB                                   KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS1.PL1LIB                                    KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS1.TELCMLIB                                  KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS2.LOADLIB                                   KEPT
IEF285I   VOL SER NOS= G1SYSG.
IEF285I   SYS1.SSPAK                                     KEPT
IEF285I   VOL SER NOS= G1SYS2.
IEF285I   SYS71137.T091913.SV000.G70RMRAY.R0000013      SYSOUT
IEF285I   VOL SER NOS= G1SCR8.
IEF285I   SYS71137.T091913.RV000.G70RMRAY.OBJMOD        PASSED
IEF285I   VOL SER NOS= 30704G.
IEF285I   SYS71137.T123456.I0007.G70RMRAY.S0000019      SYSIN
IEF285I   VOL SER NOS= G1SCR1.
IEF285I   SYS71137.T123456.I0007.G70RMRAY.S0000019      DELETED
IEF285I   VOL SER NOS= G1SCR1.
IEF285I   SYS71137.T091913.SV000.G70RMRAY.R0000014      SYSOUT
IEF285I   VOL SER NOS= G1SCR9.
IEF285I   SYS71137.T091913.SV000.G70RMRAY.R0000015      DELETED
IEF285I   VOL SER NOS= G1SCR7.
-----JOB NBR=  114 STEP NBR= 01  G70RMRAY  GO        PGM=LOADER   CARDS=00013   INITIATION TIME=09.21.06.21 DATE=05-17-71
----- CPU=000.2 I/O=000.1 CORE=000.4  CHARGE=000.22         STEP=01  GO         TERMINATION TIME=09.29.38.26 DATE=05-17-71
----- I/O TIME BY DEVICE.  DISK=*****2.00,DRUM=******.00,TAPE=******.14,CELL=******.00,OTHR=******.00
----- STEP REGION SIZE=0400K  MAXIMUM REGION SIZE USED=0400K   PERCENT OF REGION USED=99
IEF285I   SYS71137.T091913.RV000.G70RMRAY.OBJMOD        KEPT
IEF285I   VOL SER NOS= 30704G.


----- CPU=000.2 I/O=000.1 CORE=000.4  CHARGE=000.22    JOB NBR=114  G70RMRAY  360/95      SYSTEM=MVT-19.6   01-16-71  G1
----- I/O TIME BY DEVICE.  DISK=*****2.00,DRUM=******.00,TAPE=******.14,CELL=******.00,OTHR=******.00
```

METHOD OF ANALYSIS

The following parameters used in the analysis are input variables. All other parameters are built-in constants, or are calculated from the input data:

Δt_c	Coast interval
Δt_b	Burn interval
W_0	Initial weight
ΔW	Weight change during burn
α	Nominal pitch during burn
ψ	Nominal yaw during burn
T	Nominal thrust during burn
$\vec{X}$	Nominal initial state vector (X_{0j})
P_0	Initial state covariance matrix
σ_T, σ_α, σ_Y	Errors in thrust, pitch, yaw

The initial state covariance matrix can be input in local tangent coordinates, but all computation is made in an inertial central body Cartesian framework. This coordinate system applies to the analysis presented. (See Figure 3.)

Propagation of the State Vector and Covariance Matrix Through Coast

This is accomplished by integrating the equations of motion for the two-body problem through the coast time interval. Let t_0 be the initial time. We define t_1, $\vec{X}_1$, P_1 to be the time, state vector, and state covariance matrix at the end of coast ($t_1 = t_0 + \Delta t_c$).

The equations of motion are:

$$\ddot{\vec{r}} = -\mu \, \vec{r} / r^3 \, ,$$

where $\vec{r}$ (t) represents the position vector at time t, and μ is the gravitational constant times mass of central body.

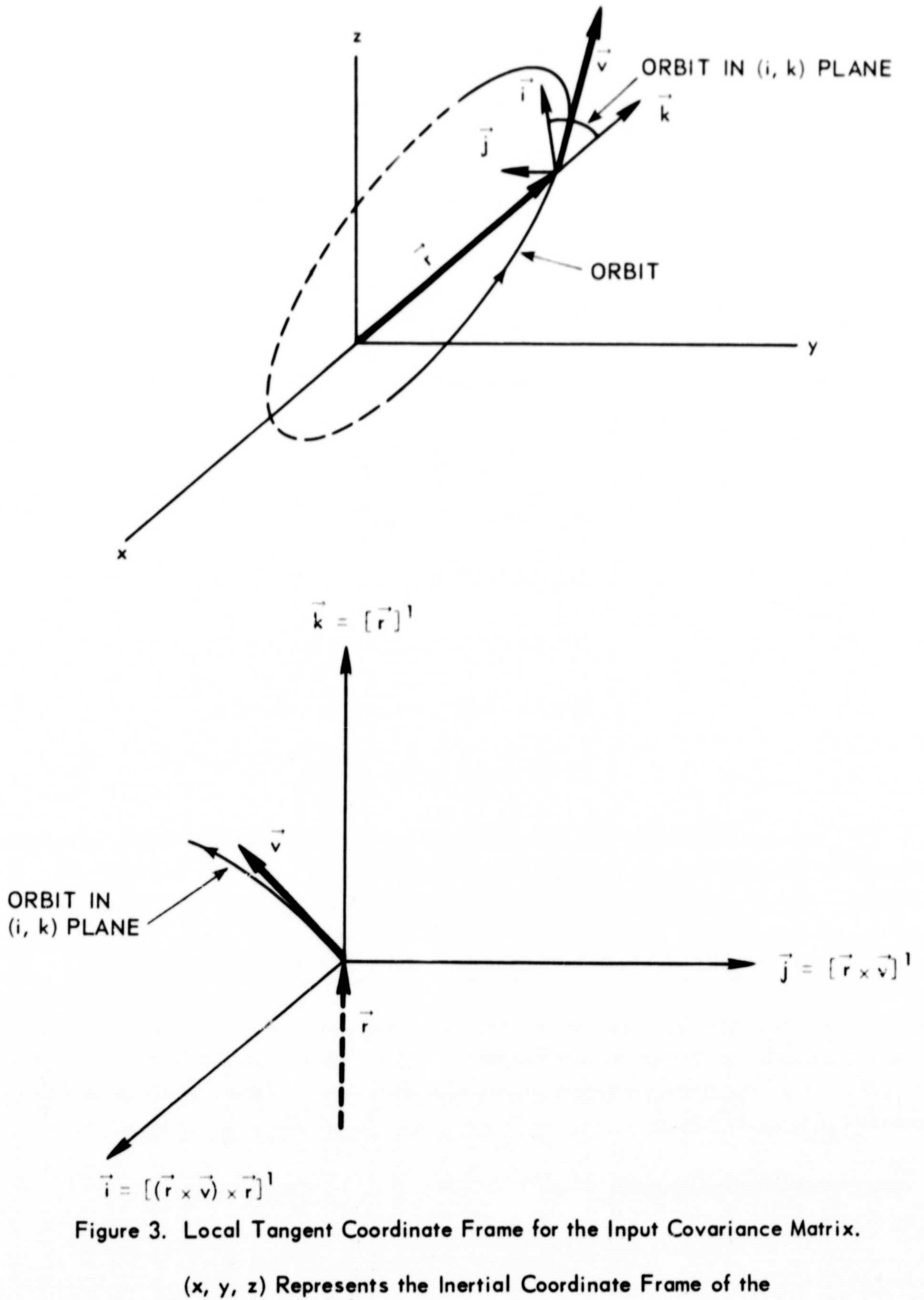


Figure 3. Local Tangent Coordinate Frame for the Input Covariance Matrix.

(x, y, z) Represents the Inertial Coordinate Frame of the

State Vector $\begin{pmatrix} \vec{r} \\ \vec{v} \end{pmatrix}$.

(i, j, k) Represents the Local Tangent Frame.

The integration is accomplished by making use of the subroutine TWOBDY prepared by W. H. Goodyear.* Goodyear defines a new variable φ by the differential equation:

$$\dot{\varphi} = 1/r ,$$

where φ is zero when t is t_0. Then $\vec{r}$ and $\dot{\vec{r}}$ are expanded in a Taylor series about $\vec{r}\ (t_0)$ in terms of φ. From these series expansions it is possible to obtain the state vector at any given time. In this way $\vec{X}_1 = \begin{pmatrix} \vec{r}\ (t_1) \\ \dot{\vec{r}}\ (t_1) \end{pmatrix} = (x_{1i})$ is evaluated, and the state transition matrix $\Phi_1 = \left(\frac{\partial x_{1i}}{\partial x_{0j}} \right)$ is found by computing the partial derivatives of the terms in the Taylor series.

The covariance matrix P_1 is then evaluated by:

$$P_1 = \Phi_1 P_0 \Phi_1^T$$

Propagation of the State Vector and Covariance Matrix Through Burn, without Addition of Burn Errors

Let t_2, X_2, P_2 represent the time, state vector, and state covariance matrix (without burn errors) after burn ($t_2 = t_1 + \Delta t_b$).

The equations of motion are:

$$\ddot{\vec{r}} = - \mu \vec{r} / r^3 + \frac{\vec{\lambda}}{m} ,$$

where m(t) is the mass of the vehicle and $\vec{\lambda}$ (t) represents the thrust vector, evaluated by:

*Goodyear, W. H., "A General Method for the Computation of Cartesian Coordinates and Partial Derivatives of the Two-Body Problem," NASA CR-522.

$$\vec{\lambda} = \left(\frac{\dot{\vec{r}}}{\dot{r}}, \; \frac{(\vec{r} \times \dot{\vec{r}}) \times \dot{\vec{r}}}{\left\| (\vec{r} \times \dot{\vec{r}}) \times \dot{\vec{r}} \right\|}, \; \frac{\vec{r} \times \dot{\vec{r}}}{\left\| \vec{r} \times \dot{\vec{r}} \right\|} \right) \begin{pmatrix} T \cos\psi \cos\alpha \\ -T \cos\psi \sin\alpha \\ T \sin\psi \end{pmatrix}$$

An approximate solution to the equations of motion is obtained by integrating:

$$\ddot{\vec{r}} = -\frac{\mu \vec{r}_1}{r_1^3} + \frac{\vec{\lambda}_1}{m},$$

where $\vec{\lambda}_1 = \vec{\lambda}(t_1)$, and $\vec{r}_1 = \vec{r}(t_1)$.

The solution for this equation through burn is:

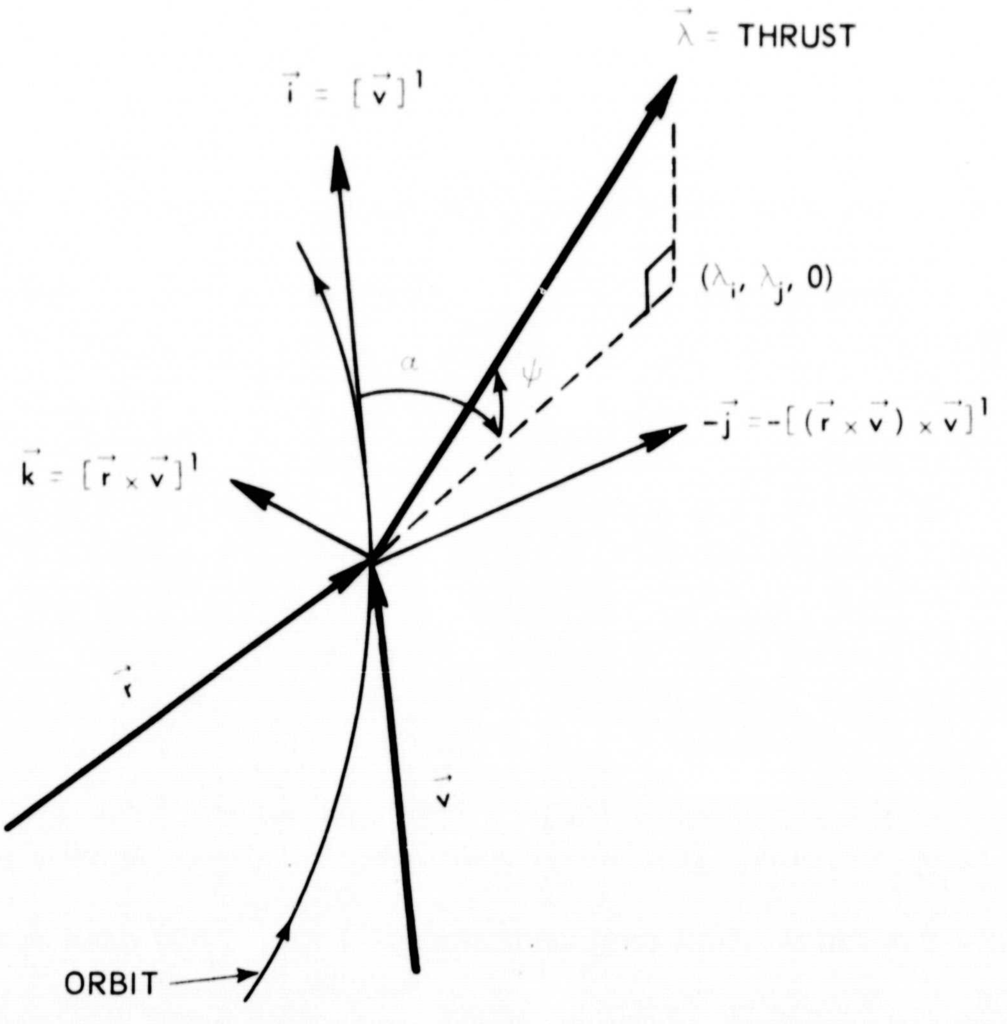
$$\dot{\vec{r}}_2 = \dot{\vec{r}}_1 - \vec{r}_1 \frac{\mu \Delta t_b}{r_1^3} + \frac{\vec{\lambda}_1 \Delta t_b}{\Delta m} \log \left(\frac{m_1 + \Delta m}{m_1} \right)$$

$$\vec{r}_2 = \vec{r}_1 + \dot{\vec{r}}_1 \Delta t_b - \vec{r}_1 \frac{\mu \Delta t_b^2}{2 r_1^3} + \vec{\lambda}_1 \frac{\Delta t_b^2}{\Delta m} \left[\frac{m_1 + \Delta m}{\Delta m} \log \frac{(m_1 + \Delta m)}{m_1} - 1 \right]$$

where $\vec{r}_1$, m_1 are the position vector and mass before burn, Δm is the mass change during burn, and $\vec{r}_2$ is the position vector after burn. This provides the state vector $\vec{X}_2 = \begin{pmatrix} \vec{r}_2 \\ \dot{\vec{r}}_2 \end{pmatrix} = (x_{2i})$. The state transition matrix $\Phi_2 = \left(\frac{\partial x_{2i}}{\partial x_{1j}} \right)$ is found by computing the partial derivatives from the above equations, and the covariance matrix P_2 is evaluated as before:

$$P_2 = \Phi_2 P_1 \Phi_2^T .$$

For long burn maneuvers provision is made to reduce the error resulting from the approximate solution to the equations of motion. An input parameter is available which breaks up the burn interval into n_b sub-intervals; the state vector and covariance matrix are then propagated as above through each burn



$\vec{\lambda}$ = THRUST
$\vec{i} = [\vec{v}]^1$
$(\lambda_i, \lambda_j, 0)$
α
ψ
$-\vec{j} = -[(\vec{r} \times \vec{v}) \times \vec{v}]^1$
$\vec{k} = [\vec{r} \times \vec{v}]^1$
$\vec{r}$
$\vec{v}$
ORBIT

Figure 4. Pitch and Yaw Angles.
The Pitch, α, is Measured Toward $-\vec{j}$ in the (i, j) Plane (Orbit Plane) from $\vec{i}$ to the Projection, $(\lambda_i, \lambda_j, 0)$ of the Thrust Vector, $\vec{\lambda}$. Yaw, ψ, is Measured Toward $\vec{k}$ from $(\lambda_i, \lambda_j, 0)$ to $\vec{\lambda}$.

step. Variations in the thrust vector through burn can easily be programmed into this scheme, if step functions are assumed. The computation of P_2 procedes as follows:

$$\Phi_2^T = \prod_{i=1}^{n_b} \Phi_{2_i}^T$$

$$P_2 = \Phi_2 \, P_1 \, \Phi_2^T$$

<u>The Inclusion of the Burn Errors into the Covariance Matrix P_2</u>

Using the input errors σ_T, σ_α, σ_ψ a diagonal covariance matrix for burn errors is assumed:

$$B = \begin{pmatrix} \sigma_T^2 & 0 & 0 \\ 0 & \sigma_\alpha^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{pmatrix}$$

This is transformed into a thrust covariance matrix first in a local tangent Cartesian coordinate frame and then in the reference inertial coordinate system. Finally the state-thrust transition matrix $\left(\frac{\partial x_{2i}}{\partial \lambda_{1j}} \right)$ is used to evaluate the state covariance matrix due to burn, and this is added to P_2 to obtain the final state covariance matrix, $\tilde{P}_2$, after burn. Let

$$U = \begin{pmatrix} \cos\alpha \cos\psi & -T \sin\alpha \cos\psi & -T \cos\alpha \sin\psi \\ -\sin\alpha \cos\psi & -T \cos\alpha \cos\psi & T \sin\alpha \sin\psi \\ \sin\psi & 0 & T \cos\psi \end{pmatrix}$$

$$T = \left(\frac{\dot{\vec{r}}_1}{\dot{r}_1} , \frac{(\vec{r}_1 \times \dot{\vec{r}}_1) \times \dot{\vec{r}}}{\left\| (\vec{r}_1 \times \dot{\vec{r}}_1) \times \dot{\vec{r}}_1 \right\|} , \frac{\vec{r}_1 \times \dot{\vec{r}}_1}{\left\| \vec{r}_1 \times \dot{\vec{r}}_1 \right\|} \right) ,$$

and

$$Q = \left(\frac{\partial x_{2i}}{\partial \lambda_{1j}} \right) ,$$

where $\vec{\lambda}_1 = (\lambda_{1i})$ is the thrust vector before burn.

The equations of Q are:

$$\frac{\partial r_{2i}}{\partial \lambda_{1j}} = \frac{\lambda_{1i} \lambda_{1j} \Delta t_b^2}{\lambda_1^2 \Delta m} \left[2 - \left(\frac{2m_1 + \Delta m}{\Delta m} \right) \log \left(\frac{m_1 + \Delta m}{m_1} \right) \right]$$

$$+ \delta_{ij} \frac{\Delta t_b^2}{\Delta m} \left[\frac{m_1 + \Delta m}{\Delta m} \log \left(\frac{m_1 + \Delta m}{m_1} \right) - 1 \right] ,$$

$$\frac{\partial \dot{r}_{2i}}{\partial \lambda_{1j}} = \frac{\lambda_{1i} \lambda_{1j}}{\lambda_1^2} \frac{\Delta t_b}{\Delta m} \left[\frac{\Delta m}{m_1 + \Delta m} - \log \left(\frac{m_1 + \Delta m}{m_1} \right) \right] + \delta_{ij} \frac{\Delta t_b}{\Delta m} \log \left(\frac{m_1 + \Delta m}{m_1} \right) ,$$

where $\delta_{ij} = 0, i \neq j$ and $\delta_{ii} = 1$.

The final covariance matrix after burn is computed as follows:

$$\tilde{P}_2 = P_2 + Q\, T\, U\, B\, U^T\, T^T\, Q^T .$$

If the burn interval is divided into n_b sub-intervals by the input option, the equations for the covariance propagation become:

$$\theta = \sum_{k=1}^{n_b - 1} \prod_{i=n_b}^{k+1} \Phi_i\, Q_k\, T_k\, U_k + Q_{n_b}\, T_{n_b}\, U_{n_b}$$

$$\tilde{P}_2 = P_2 + \theta\, B\, \theta^T$$

Computation of Histograms of State Dependent-Variables

Let $\xi(\vec{X})$ represent the vector of state dependent-variables for which histograms are desired. The distribution of $\xi(\vec{X}_2)$ is computed as follows:

a. The Eigenvalues (E_i) and associated matrix of Eigenvectors, R, is first computed for $\tilde{P}_2$:

$$\text{diag}\,(E_i) = R^{-1}\,\tilde{P}_2\,R.$$

b. A set of n normally distributed random vectors, $\{\Delta\vec{\beta}_i : i = 1, \ldots, n\}$, is generated with mean (0) and variance (E_i).

c. A reverse transformation is made on each random vector, and the mean is shifted to the nominal by addition of X_2:

$$\vec{\gamma}_i = \vec{X}_2 + R\,\Delta\vec{\beta}_i, \quad i = 1, \ldots, n.$$

d. Finally $\{\xi(\vec{\gamma}_i) : i = 1, \ldots, n\}$ is computed, and the distributions of the state dependent-variables are presented as histograms of the elements of the vectors $\{\xi(\vec{\gamma}_i)\}$.

Computation of the Delta-Velocity required to effect a given Semi-Major Axis, Eccentricity and Inclination

To change semi-major axis and eccentricity, thrusting is assumed to be at apoapsis and periapsis. To change inclination thrusting is assumed to be at one of the nodes. If no inclination is desired, δv is calculated for both apoapsis-periapsis and periapsis-apoapsis burn sequences. If an inclination change is desired, twelve possible burn sequences of apoapsis-periapsis-node are considered. The minimum of all possible δv calculations is accepted. The following identities are used to calculate the δv's in the burn sequences:

$$r_a = a(1 + e)$$

$$r_p = a(1 - e)$$

$$v = \mu\left(\frac{2}{r} - \frac{1}{a}\right)$$

$$r_n = \frac{r_p\, r_a}{a}\left[1 \pm \left(\frac{r_a}{a} - 1\right) \cos w\right]^{-1}$$

$$\delta v_i = v(\delta i)$$

where:

r_a = Apoapsis radius

r_p = Periapsis radius

a = Semi-major axis

e = Eccentricity

v = Velocity

μ = Gravitational constant times mass of central body

r = Radius

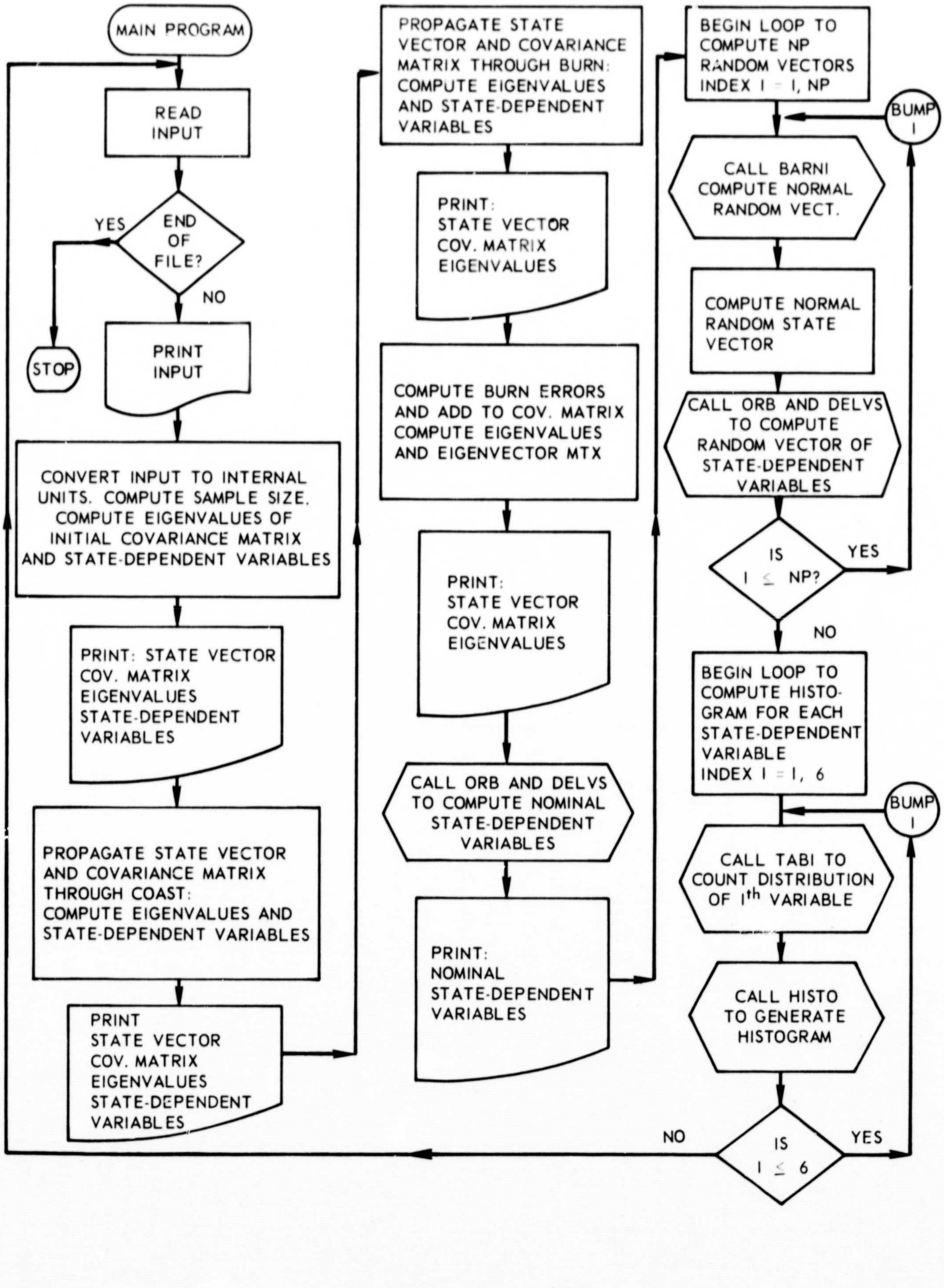
r_n = Radius at node

w = Argument of periapsis

δv_i = Velocity to change inclination

δi = Desired inclination change

FLOWCHART



MAIN PROGRAM
READ INPUT
END OF FILE?
YES
NO
STOP
PRINT INPUT
CONVERT INPUT TO INTERNAL UNITS. COMPUTE SAMPLE SIZE. COMPUTE EIGENVALUES OF INITIAL COVARIANCE MATRIX AND STATE-DEPENDENT VARIABLES
PRINT: STATE VECTOR COV. MATRIX EIGENVALUES STATE-DEPENDENT VARIABLES
PROPAGATE STATE VECTOR AND COVARIANCE MATRIX THROUGH COAST: COMPUTE EIGENVALUES AND STATE-DEPENDENT VARIABLES
PRINT STATE VECTOR COV. MATRIX EIGENVALUES STATE-DEPENDENT VARIABLES
PROPAGATE STATE VECTOR AND COVARIANCE MATRIX THROUGH BURN: COMPUTE EIGENVALUES AND STATE-DEPENDENT VARIABLES
PRINT: STATE VECTOR COV. MATRIX EIGENVALUES
COMPUTE BURN ERRORS AND ADD TO COV. MATRIX COMPUTE EIGENVALUES AND EIGENVECTOR MTX
PRINT: STATE VECTOR COV. MATRIX EIGENVALUES
CALL ORB AND DELVS TO COMPUTE NOMINAL STATE-DEPENDENT VARIABLES
PRINT: NOMINAL STATE-DEPENDENT VARIABLES
BEGIN LOOP TO COMPUTE NP RANDOM VECTORS INDEX I = I, NP
BUMP I
CALL BARNI COMPUTE NORMAL RANDOM VECT.
COMPUTE NORMAL RANDOM STATE VECTOR
CALL ORB AND DELVS TO COMPUTE RANDOM VECTOR OF STATE-DEPENDENT VARIABLES
IS I ≤ NP?
YES
NO
BEGIN LOOP TO COMPUTE HISTOGRAM FOR EACH STATE-DEPENDENT VARIABLE INDEX I = I, 6
BUMP I
CALL TABI TO COUNT DISTRIBUTION OF Ith VARIABLE
CALL HISTO TO GENERATE HISTOGRAM
IS I ≤ 6
NO
YES

LISTING

The following is a FORTRAN source listing of the entire program.

The listing is arranged as follows:

LISTING (Continued)

```
C MAIN PROGRAM
C
C THERR -- TWO BODY ERROR ANALYSIS PROGRAM
C
      IMPLICIT REAL*8(A-H,O-Y)
      REAL*8 NAMES
      DIMENSION SIGB(3),XI(6),PI(6,6),C1(6,6),C2(9),C3(6),C4(3),C5(3)
      DIMENSION XC(6),P(6,6),PX(6,6),Q(6,6),R(6,6),PAR(3,3),T50(6,3)
      DIMENSION ZPCT(51),ZSTATS(5),TITLE(10),ELT(9)
      DIMENSION U(6,6),B(3,3),E(6),EV(6,6),ZB(10000,7)
      DIMENSION ZUB(3,6),ZFREQ(50)
      DIMENSION TV(3),ZX(50),XP(6),A(9)
      DIMENSION ZELT(6)
      DIMENSION NAMES(6)
      DIMENSION V(6,6)
      DIMENSION H(6,6)
      DIMENSION IHIST(6),XII(6),PII(21),SIGBI(3)
      DATA ELT/3HECC,3HSMA,3HINC,5HOMEGA,4HARGP,5HTHETA,3HPER,4HAPOG,
     2 4HDELV/
      DATA NAMES/3HECC,3HSMA,3HINC,5HOMEGA,4HARGP,5HTHETA/
      NAMELIST /TBDATA/DTCI,DTBI,WI,WCI,PITCHI,YAWI,THRUSI ,ZDELT,NCONF,
     2 AS,ES,AIS,NBURN,SIGBI,ZUB,XII,IHIST,PII,XMU,IPCOOR                  ****
      CALL ERRSET (210,256,-1,1,1)
      DTCI = 0.D0
      DTBI = 0.D0
      WI = 0.D0
      WCI = 0.D0
      PITCHI = 0.D0
      YAWI = 0.D0
      THRUSI = 0.D0
      ZDELT = 0.
      NCONF = 0
      AS = 0.D0                                                            ****
      ES = 0.D0                                                            ****
      IPCOOR = 1                                                           ****
      AIS = -1.D0
      NBURN = 1
      SIGBI(1) = 0.D0
      SIGBI(2) = 0.D0
      SIGBI(3) = 0.D0
      DO 895 I = 1,6
      ZUB(1,I) = 0.
      ZUB(2,I) =50.
      ZUB(3,I) = 0.
      XII(I) = 0.D0
      IHIST(I) = I
  895 CONTINUE
      DO 899 I = 1,21
      PII(I) = 0.D0
  899 CONTINUE
      FACTOR = .0044482217D0/.00980665D0
      XMU = 398603.2D0
   10 READ (5,500,END=9999) (TITLE(I),I=1,10)
      READ (5,TBDATA)
      DTC = DTCI
      DTB = DTBI
      W = WI
      WC = WCI
      PITCH = PITCHI
      YAW = YAWI
      THRUST = THRUSI
      SIGB(1) = SIGBI(1)
      SIGB(2) = SIGBI(2)
      SIGB(3) = SIGBI(3)
      DO 897 I = 1,6
      XI(I) = XII(I)
      DO 896 J = I,6
      L = I*(13-I)/2+J-6
      PI(I,J) = PII(L)
      PI(J,I) = PI(I,J)
  896 CONTINUE
  897 CONTINUE
      NP = NCONF
      IF (NP.LE.0) GO TO 15
      IF (NP.LE.6) NP = NSAMP(NCONF,ZDELT)
      IF (NP.GT.10000) NP = 10000
   15 CONTINUE
      WRITE(6,600) (TITLE(I),I=1,10)
      WRITE (6,610) DTC,DTB,W,WC,PITCH,YAW,THRUST,ZDELT,NCONF,IHIST
      WRITE (6,605) AS,ES,AIS,NBURN,XMU                                 79000    ****
      WRITE (6,615) ((ZUB(I,J),J=1,6),I=1,3)                            80000
      WRITE (6,620) (SIGB(I),I=1,3)                                     81000
      WRITE (6,630) (XI(I),I=1,6)                                       82000
      WRITE (6,640) IPCOOR,((PI(I,J),J=1,6),I=1,6)                      83000    ****
      ENBURN = NBURN                                                    83100
      DTB = DTB/ENBURN                                                  83200
      WC = WC/FNBURN                                                    83210
      AIS = AIS/57.29577951D0                                           84000
      CALL ORB(XI,XI(4),XMU,C2)                                         85000
                                                                                 ****DELETIONS
      DO 18 I = 1,6                                                     87000
      XC(I) = XI(I)                                                     88000
      XB(I) = XI(I)                                                     89000
      DO 17 J = 1,6                                                     90000
      IF (IPCOOR.LE.2) PI(I,J) = PI(I,J)*.0003048**2                    91000    ****
      P(I,J) = PI(I,J)                                                  92000
      V(I,J) = 0.D0                                                     93000
      H(I,J) = 0.D0                                                     93100
   17 CONTINUE                                                          94000
      H(I,I) = 1.D0                                                     94100
   18 CONTINUE                                                          95000
      IF (IPCOOR.EQ.1.OR.IPCOOR.EQ.3) CALL CONVET(PI,XI,XI(4),2,PI)     96000    ****
      WRITE (6,650)                                                     97000
      CALL MTXPR(XI,PI,E,EV)                                            98000
      WRITE (6,700) (NAMES(I),I=1,6),(C2(I),I=1,6)                      99000
      PSI = 0.D0                                                       100000
      IF (DTC.EQ.0.D0) GO TO 551                                       101000
      CALL TWOBDY(XI,DTC,XMU,PSI ,XC,P,C1,C2,C3,C4,C5,C6.C7)           102000
      WRITE (6,655) ((P(I,J),J=1,6),I=1,6)                             103000
      CALL MTRX(P,PI,P,6,6,-1)                                         104000
      WRITE (6,660)                                                    105000
      CALL MTXPR(XC,P,E,EV)                                            106000
      CALL ORB(XC,XC(4),XMU,C2)                                        107000
      WRITE (6,700) (NAMES(I),I=1,6),(C2(I),I=1,6)                     108000
  551 IF (DTB.EQ.0.D0) GO TO 552                                       109000
      DO 886 J=1,6                                                     110000
      C2(J)=XC(J)                                                      111000
  886 CONTINUE                                                         112000
      SIGB(1) = (SIGB(1)*.004448221700)**2                             113000
      SIGB(2) = (SIGB(2)*.01745329300)**2                              114000
      SIGB(3) = (SIGB(3)*.01745329300)**2                              115000

      DO 30 I = 1,3                                                    116000
      DO 20 J = 1,3                                                    117000
      B(I,J) = 0.D0                                                    118000
   20 CONTINUE                                                         119000
      B(I,I) = SIGB(I)                                                 120000
   30 CONTINUE                                                         121000
      DO 888 I=1,NBURN                                                 122000
      CALL BURNST(C2,C2(4),XB,XB(4),W,0.D0,WC,PITCH,TV,DTB,THRUST,YAW, 123000
     2 XMU)                                                            124000
      XMASS = W*FACTOR                                                 125000
      XMDOT = WC/DTB*FACTOR                                            126000
      CALL POWERX(C2,C2(4),TV,XMASS,XMDOT,0.D0,XMU,DTB,O,R)            127000
      CALL MTRX(O,H,H,6,6,0)                                           128000
      CALL MTRX(O,V,V,6,3,0)                                           128100
      CALL PARTAL (PAR,C2,C2(4),T50,PITCH,YAW,THRUST)                  129000
      CALL MTRPLY (T50,PAR,U,3,3,3,6,3,6)                              130000
      CALL MTRPLY (R,U,T50,6,3,3,6,6,6)                                131000
      DO 676 J = 1,6                                                   132000
      DO 677 L = 1,3                                                   132100
      V(J,L) = V(J,L)+T50(J,L)                                         132200
  677 CONTINUE                                                         132300
  676 CONTINUE                                                         132400
C-------------------                                                   133000
C                                                                      134000
C         INSERT BURN CORRECTIONS HERE WHEN THEY'RE READY . ..         135000
      IF (YAWI.EQ.0.D0) GU TO 578                                      136000
      V1DV2=C2(4)*XB(4)+C2(5)*XB(5)+C2(6)*XB(6)                        137000
      ABSV1=DSQRT(C2(4)**2+C2(5)**2+C2(6)**2)                          138000
      ABSV2=DSQRT(XB(4)**2+XB(5)**2+XB(6)**2)                          139000
      BETA=DARCOS(V1DV2/(ABSV1*ABSV2))                                 140000
      YAW=YAW+BETA/.01745329 3D0                                       141000
  578 CONTINUE                                                         142000
C                                                                      143000
C-------------------                                                   144000
      DO 887 J=1,6                                                     145000
      C2(J)=XB(J)                                                      146000
  887 CONTINUE
      W=W+WC
  888 CONTINUE
      CALL MTRX(H,P,P,6,6,-1)
      CALL MTRX(V,B,V,3,3,-1)
      WRITE (6,670)
      CALL MTXPR(XB,P,E,EV)
      WRITE (6,680)
      CALL MTXPR(XB,V,E,EV)
      DO 50 I = 1,6
      DO 40 J = 1,6
      P(I,J) = P(I,J)+V(I,J)
   40 CONTINUE
   50 CONTINUE
      WRITE (6,690)
      CALL MTXPR(XB,P,E,EV)
      CALL ORB(XB,XB(4),XMU,C2)
      WRITE (6,700) (NAMES(I),I=1,6),(C2(I),I=1,6)
  552 CONTINUE
      AI = C2(3)/57.29577951D0
      APF = C2(5)/57.2957795 1D0
      C2(8) = C2(2)*(1.D0+C2(1))
      C2(7) = C2(2)*(1.D0-C2(1))
      C2(9) = 0.D0
      IF (AS.GT.0.D0) CALL DELVS(C2(2),C2(8),C2(7),AS,ES,AI,AIS,APF,XMU,        ****
     2 C2(9))
      WRITE (6,740) (ELT(I),I=7,9),(C2(I),I=7,9)
      IF (NP.LE.0) GO TO 10
      DO 200 J = 1,NP
      DO 150 I = 1,6
      SD = 0.D0
      IF (E(I).GT.0.D0) SD=DSQRT(E(I))
      A(I) = BARN1(-1,-1,12787,SD)
  150 CONTINUE
      CALL MTRPLY(EV,A,XC,6,6,1,6,6,6)
      DO 160 I = 1,6
      XC(I) = XB(I)+XC(I)
  160 CONTINUE
      CALL ORB(XC,XC(4),XMU,A)
      AI = A (3)/57.2957795lD0
      APF = A(5)/57.2957795lD0
      A(8) = A(2)*(1.D0+A(1))
      A(7) = A(2)*(1.D0-A(1))
      A(9) = 0.D0
      IF(AS.GT.0.D0)CALL DELVS(A(2),A(8),A(7),AS,ES,AI,AIS,APF,XMU,A(9))        ****
      DO 180 I = 1,6
      L = IHIST(I)
      IF (L.LE.0.OR.L.GT.9) GO TO 180
      ZB(J,I) = A(L)
      ZB(J,7) = 1.
  180 CONTINUE
  200 CONTINUE
      DO 250 I = 1,6
      L = IHIST(I)
      IF (L.LE.0.OR.L.GT.9) GO TO 250
      ZELT(I) = C2(L)
      NZP = ZUB(2,I)
      NZX = NZP-1
      CALL TAB1(ZB(1,I),ZB(1,7),1,ZUB(1,I),ZFREQ,ZPCT,ZSTATS,NP,1)
      IF (ZUB(3,I).GT.ZUB(1,I)) GO TO 220
      ZUB(1,I) = ZSTATS(4)
      ZUB(3,I) = ZSTATS(5)
  220 ZDX = (ZUB(3,I)-ZUB(1,I))/(ZUB(2,I)-2.)
      ZX(1) = ZUB(1,I)
      DO 230 J = 2,NZX
      ZX(J) = ZX(J-1)+ZDX
  230 CONTINUE
      CALL HISTO(TITLE,ELT(L),ZPCT,ZX,ZSTATS(2),ZSTATS(3),NZP,ZELT(I),
     1       NP)
  250 CONTINUE
      GO TO 10
 9999 CONTINUE
      STOP
  500 FORMAT (10A8)
  600 FORMAT (1H1,4X,10A8)
  605 FORMAT (/23X,2HAS,23X,2HES,22X,3HAIS,20X,5HNBURN,22X,3HXMU/               ****
     2 3D25.16,I25,D25.16)                                                      ****
  610 FORMAT (/22X,3HDTC,22X,3HDTB,24X,1HW,23X,2HWC,20X,5HPITCH/
     1 5D25.16//22X,3HYAW,19X,6HTHRUST,20X,5HZDELT,20X,5HNCONF,
    2 20X,5HIHIST/2D25.16,E25.8,I25,13X,6I2)
  615 FORMAT (/5X,22HSAMPLE INTERVAL MATRIX/(   6F20.8))
  620 FORMAT (/5X,4HSIGB/3D25.16)
  630 FORMAT (/5X,20HINITIAL STATE VECTOR/6D21.13)
  640 FORMAT (/5X,25HINITIAL COVARIANCE MATRIX,5X,8HIPCOOR =,I2/(6D21.13      ****
    2 ))                                                                      ****
  650 FORMAT (//25X,12HBEFORE COAST//)
  655 FORMAT (/5X,17HTRANSITION MATRIX/(6D21.13))
  660 FORMAT (//25X,11HAFTER COAST//)
  670 FORMAT (//25X,10HAFTER BURN//)
  680 FORMAT (//25X,11HBURN ERRORS//)
  690 FORMAT (//25X,27HAFTER BURN WITH BURN ERRORS//)
  700 FORMAT (/5X,24HORBITAL ELEMENTS NOMINAL/6(13X,A8)/6D21.13)
  710 FORMAT (//5X,A8,5X,6HMEAN =,F20.8,5X,6HSDEV =,F20.8,5X,4HNP =,I6//
    1 5X,34HINTERVALS AND PERCENTAGE FREQUENCY//5X,' MINUS INF. ',
    2 1P2E12.4)
  720 FORMAT (5X,1P3E12.4)
  730 FORMAT (5X,1PE12.4,' PLUS INF.  ', 1PE12.4)
  740 FORMAT (/3(13X,A8)/3D21.13)
      END

C
      SUBROUTINE MTXPR(X,P,E,EV)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(5),P(6,6),E(6),EV(6,6),PX(6,5)
      DIMENSION B(6,6)
      CALL CONVET (P,X,X(4),0,PX)
      DO 50 I = 1,6
      DO 40 J = 1,6
      EV(I,J) = P(I,J)
      B(I,J) =PX(I,J)/.0003048**2
   40 CONTINUE
   50 CONTINUE
      CALL EIGEN(EV,E,6,1)
      WRITE (6,600) (X(I),I=1,6)
      WRITE (6,610) ((PX(I,J),J=1,6),I=1,6)
      WRITE (6,640) ((B(I,J),J=1,6),I=1,6)
      WRITE (6,620) ((P (I,J),J=1,6),I=1,6)
      WRITE (6,630) (E(I),I=1,6)
      RETURN
  600 FORMAT (/5X,12HSTATE VECTOR/(6D21.13))
  610 FORMAT (/5X,33HCOVARIANCE MATRIX (LOCAL TANGENT)/(6D21.13))
  620 FORMAT (/5X,28HCOVARIANCE MATRIX (INERTIAL)/(6D21.13))
  630 FORMAT (5X,11HEIGENVALUES/(6D21.13))
  640 FORMAT (/5X,33HCOVARIANCE MATRIX (LOC TAN -- FT)/(6D21.13))
      END

C
      FUNCTION NSAMP(M,DELTA)
C
C     M SETS CONFIDENCE LEVEL
C     IF M=1,LEVEL IS .9
C     IF M=2, LEVEL IS .95
C     IF M=3, LEVEL IS .98
C     IF M=4, LEVEL IS .99
C     IF M=5, LEVEL IS.995
C     IF M=6, LEVEL IS.999
C     DELTA SETS UNCERTAINTY
      IF (M-1) 3,2,3
    2 A=1.282
      GO TO 13
    3 IF (M-2) 5,4,5
    4 A=1.645
      GO TO 13
    5 IF(M-3) 7,6,7
    6 A=2.054
      GO TO 13
    7 IF (M-4) 9,8,9
    8 A=2.326
      GO TO 13
    9 IF(M-5) 11,10,11
   10 A=2.576
      GO TO 13
   11 A=3.09
   13 NSAMP=(A/(2.*DELTA))*(A/(2.*DELTA))
      RETURN
      END
```

```
C
      REAL FUNCTION BARN1*8(I,IKEY,IFRN,SD)
C
      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C       SD------- THE DESIRED STANDARD DEVIATION
C       AMEAN--- THE DESIRED MEAN
C       H--------- THE POPULATION SIZE
      DATA AMEAN/0.D0/
      DATA IHEFE/12787/
      DATA H/36.D0/
      IF (IKEY)5,4,4
    4 IHERE=IFRN
    5 IF(I)6,7,7
    6 CALL GAUSS(IHERE,SD,AMEAN,VAL,H)
      IFRN=IHEFE
      GO TO 8
    7 CALL RANDU(IHERE,IFRN,VAL)
      IHERE=IFRN
    8 BARN1=VAL
      RETURN
      END


C
      SUBROUTINE GAUSS(/IX/,S,AM,V,H)
C
      IMPLICIT REAL*8(A-H,O-Z)
      K=H
      A=0.0D0
      DO 50 I=1,K
      CALL RANDU(IX,IY,Y)
      IX=IY
   50 A=A+Y
      HO=H/12.
      H2=H/2.
      V=(S*(A-H2))/DSQRT(HO)+AM
      RETURN
      END


C
      SUBROUTINE RANDU(/IX/,IY,YFL)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DATA JJJ5/1027/
      IY=IX*JJJ5
      IF(IY) 5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*.4656613D-9
      RETURN
      END


C
      SUBROUTINE EIGEN(AA,VALU,NR,M)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 IND
C     EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC MATRIX
C
C
      DIMENSION A(8,8),B(8,8),VALU(8),DIAG(8),SUPERD(7),Q(7),VALL(8)
     1,S(7),C(7),D(8),IND(8),U(8),DUMMY(94),AA(64)
      EQUIVALENCE (DIAG(1),DUMMY(1)),(SUPERD(1),DUMMY(9)),
     1(VALL(1),D(1),DUMMY(16)),(Q(1),S(1),DUMMY(24)),(B(1,1),DUMMY( 31)),
     2 (IND(1),U(1)),(II,MATCH),(TAU,BETA),(P,PRODS),(T,SMALLD),
     3 (ANORM,ANORM2)
      SQRT(X)=DSQRT(X)
      SIN(Y)=DSIN(Y)
      COS(Z)=DCOS(Z)
      ABS(A)=DABS(A)
C
C     CALCULATE NORM OF MATRIX
C
      N=NR
      ORMA = 0.D0
      J =1
      DO 1 I=1,N
      ORMA = ORMA+AA(J)
1     J=J+N+1
      DO 2 I=1,N
      NI=N*(I-1)
      DO 2 J=1,N
      IJ=NI+J
2     A(J,I) = AA(IJ)/ORMA
    3 ANORM2=0.0D0
    4 DO 6 I=1,N
    5 DO 6 J=1,N
    6 ANORM2=ANORM2+A(I,J)**2
    7 ANORM=SQRT (ANORM2)
C
C     GENERATE IDENTITY MATRIX
C
    9 IF (M) 10, 45, 10
   10 DO 40 I=1,N
   12 DO 40 J=1,N
   20 IF(I-J) 35, 25, 35
   25 B(I,J)=1.0D0
   30 GO TO 40
   35 B(I,J)=0.0D0
   40 CONTINUE
C
C     PERFORM ROTATIONS TO REDUCE MATRIX TO JACOBI FORM
C
   45 IEXIT=1
   50 NN=N-2
   52 IF (NN) 890, 170, 55
   55 DO 160 I=1,NN
   60 II=I+2
   65 DO 160 J=II,N
   70 T1=A(I,I+1)
   75 T2=A(I,J)
   80 GO TO 900
   90 DO 105 K=I,N
   95 T2=CUS*A(K,I+1)+SUN*A(K,J)
  100 A(K,J)=CUS*A(K,J)-SUN*A(K,I+1)
  105 A(K,I+1)=T2
  110 DO 125 K=I,N
  115 T2=CUS*A(I+1,K)+SUN*A(J,K)
  120 A(J,K)=CUS*A(J,K)-SUN*A(I+1,K)
  125 A(I+1,K)=T2
[illegible] IF (M) 130, 160, 130
  130 DO 150 K=1,N
  135 T2=CUS*B(K,I+1)+SUN*B(K,J)
  140 B(K,J)=CUS*B(K,J)-SUN*B(K,I+1)
  150 B(K,I+1)=T2
  160 CONTINUE
C
C     MOVE JACOBI FORM ELEMENTS AND INITIALIZE EIGENVALUE BOUNDS
C
  170 DO 200 I=1,N
  180 DIAG(I)=A(I,I)
  190 VALU(I)=ANORM
  200 VALL(I)=-ANORM
  210 DO 230 I=2,N
  220 SUPERD(I-1)=A(I-1,I)
  230 Q(I-1)=(SUPERD(I-1))**2
C
C     DETERMINE SIGNS OF PRINCIPAL MINORS
C
  235 TAU=0.0D0
  240 I=1
  260 MATCH=0
  270 T2=0.0D0
  275 T1=1.0D0
  277 DO 450 J=1,N
  280 P=DIAG(J)-TAU
  290 IF(T2) 300, 330, 300
  300 IF(T1) 310, 370, 310
  310 T=P*T1-Q(J-1)*T2
  320 GO TO 410
  330 IF(T1) 335, 350, 350
  335 T1=-1.0D0
  340 T=-P
  345 GO TO 410
  350 T1=1.0D0
  355 T=P
  360 GO TO 410
  370 IF(Q(J-1)) 380, 350, 380
  380 IF(T2) 400, 390, 390
  390 T=-1.0D0
  395 GO TO 410
  400 T=1.0D0
C
C     COUNT AGREEMENTS IN SIGN
C
  410 IF(T1) 425, 420, 420
  420 IF(T) 440, 430, 430
  425 IF(T) 430, 440, 440
  430 MATCH=MATCH+1
  440 T2=T1
  450 T1=T

C     ESTABLISH TIGHTER BOUNDS ON EIGENVALUES
C
  460 DO 530 K=1,N
  465 IF (K-MATCH) 470, 470, 520
  470 IF(TAU-VALL(K)) 530, 530, 480
  480 VALL(K)=TAU
  490 GO TO 530
  520 IF(TAU-VALU(K)) 525, 530, 530
  525 VALU(K)=TAU
  530 CONTINUE
  540 IF (VALU(I)-VALL(I)-5.0D-16) 570,570,550
  550 IF(VALU(I)) 560, 580, 560
  560 IF (ABS(VALL(I)/VALU(I)-1.0D0)-5.0D-16) 570,570,580
  570 I=I+1
  575 IF(I-N) 540, 540, 590
  580 TAU=(VALL(I)+VALU(I))/2.0D0
  585 GO TO 260
C
C     JACOBI EIGENVECTORS BY ROTATIONAL TRIANGULARIZATION
C
  590 IF (M) 593, 890, 593
  593 IEXIT=2
  595 DO 610 I=1,N
  600 DO 610 J=1,N
  610 A(I,J)=0.0D0
  615 DO 850 I=1,N
  620 IF (I-1) 625, 625, 621
  621 IF (VALL(I-1)-VALU(I)-5.0D-14) 730,730,622
  622 IF (VALU(I-1)) 623, 625, 623
  623 IF (ABS(VALU(I)/VALU(I-1)-1.0D0)-5.0D-14) 730,730,625
  625 CUS=1.0D0
  628 SUN=0.0D0
  630 DO 700 J=1,N
  635 IF(J-1) 680, 680, 640
  640 GO TO 900
  650 S(J-1)=SUN
  660 C(J-1)=CUS
  670 D(J-1)=T1*CUS+T2*SUN
  680 T1=(DIAG(J)-VALU(I))*CUS-BETA*SUN
  690 T2=SUPERD(J)
  700 BETA=SUPERD(J)*CUS
  710 D(N)=T1
  720 DO 725 J=1,N
  725 IND(J)=0.0D0
  730 SMALLD=ANORM
  735 DO 780 J=1,N
  740 IF(IDINT(IND(J))-1) 750,780,780
  750 IF (ABS (SMALLD)-ABS (D(J)))780, 780, 760
  760 SMALLD=D(J)
  770 NN=J
  780 CONTINUE
  790 IND(NN)=1.0D0
  800 PRODS=1.0D0
  805 IF (NN-1) 810, 850, 810
  810 DO 840 K=2,NN
  820 II=NN+1-K
  830 A(II+1,I)=C(II)*PRODS
  840 PRODS=-PRODS*S(II)
  850 A(1,I)=PRODS
C
C     FORM MATRIX PRODUCT OF ROTATION MATRIX WITH JACOBI VECTOR MATRIX
C
  855 DO 885 J=1,N
  860 DO 865 K=1,N
  865 U(K)=A(K,J)
  870 DO 8851 I=1,N
  875 A(I,J)=0.0D0
  880 DO 8852 K=1,N
      A(I,J)=B(I,K)*U(K)+A(I,J)
 8852 CONTINUE
 8851 CONTINUE
  885 CONTINUE
      DO 886 I=1,N
      NI=N*(I-1)
      DO 886 J=1,N
      IJ=NI+J
886   AA(IJ)=A(J,I)
890   CONTINUE
      DO 891 I=1,N
891   VALU(I) = VALU(I)*ORMA
      RETURN
C
C     CALCULATE SINE AND COSINE OF ANGLE OF ROTATION
C
  900 IF (T2) 910, 940, 910
  910 T=SQRT (T1**2+T2**2)
  920 CUS=T1/T
  925 SUN=T2/T
  930 GO TO (90,650), IEXIT
  940 GO TO (160,910), IEXIT
      RETURN
      END

C
      SUBROUTINE BURNST(XTR,VTR,XT,VT,W2,DW2,WC2,ALPH,TV,DT,THRUS,YA,GMU
     2 )
C
C THIS SUBROUTINE PROPAGATES THE STATE VECTOR THRU THE BURN
C WGT IS INPUT IN      LBS           AND WILL BE DIVIDED BY G
C THRUST MUST BE IN SAME UNITS AS WGT
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XTR(3),VTR(3),XT(3),VT(3),TV(3),HTR(3),CTR(3)
      CALL CROSS(XTR,VTR,HTR)
      CALL CROSS(HTR,VTR,CTR)
      CMAG = FNORM(CTR)
      VMAG = FNORM(VTR)
      HMAG = FNORM(HTR)
10    ALPHA = ALPH *.01745329 3D0
      BETA = 1.570796327D0 +ALPHA
      COSAL = DCOS(ALPHA)
      COSBE = DCOS(BETA)
      YAW =YA      *.01745329 3D0
      SYAW = DSIN(YAW)
      CYAW = DCOS(YAW)
C        CONVERT LBS TO (KG-KM / SEC**2)
      W1 = W2 * .0044482217D0
      DW1 =DW2* .0044482217D0
      WC1=WC2 * .0044482217D0
      THRUST = THRUS * .0044482217D0
      DO 1 I = 1,3
1     TV(I) = THRUST*(COSAL*CYAW*VTR(I)/VMAG + COSBE*CYAW*CTR(I)/CMAG
     1 + SYAW*HTR(I)/HMAG)
      G = 9.80665D-3
      W = W1/G
      DW = DW1/G
      WC = WC1/(G*DT)
      TEMP = DLOG(1.D0+WC*DT/(W+DW))
      RMAG = FNORM(XTR)
      CONTINUE
      DO 2 I = 1,3
      XT(I) = XTR(I)-GMU*XTR(I)*(DT**2)/(2.D0*RMAG**3)+VTR(I)*DT+TV(I)*
     1  ((W+DW+WC*DT)/(WC**2)*TEMP-DT/WC)
      VT(I) = -GMU*XTR(I)*DT/(RMAG**3)+VTR(I)+TV(I)/WC*TEMP
    2 CONTINUE
      RETURN
      END

C
      SUBROUTINE PARTAL (PAR,TR, TV, T50,ALPH,BET,THRUS)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION PAR(3.3). T50(6.3)
      DIMENSION TR(3). TV(3). Y(3). Z(3)
      T = THRUS*.0044482217D0
      ALPHA = ALPH  /57.29577951D0
      BETA = BET    /57.29577951D0
      SALPHA = DSIN (ALPHA)
      CALPHA = DCOS(ALPHA)
      SBETA = DSIN(BETA)
      CBETA = DCOS(BETA)
      PAR(1.1) = CALPHA* CBETA
      PAR(2.1) = -SALPHA*CBETA
      PAR(3.1) = SBETA
      PAR(1.2) = -T*SALPHA*CBETA
      PAR(2.2) = -T*PAR(1.1)
      PAR(3.2) = 0.D0
      PAR(1.3) = -T*CALPHA*SBETA
      PAR(2.3) = T*SALPHA*SBETA
      PAR(3.3) = T*CBETA
      VM = FNORM(TV)
      CALL CROSS (TR.TV.Z)
      CALL CROSS (Z. TV. Y)
      YM = FNORM(Y)
      ZM = FNORM(Z)
      DO 1 I=1.3
      T50(I.1) = TV(I)/VM
      T50(I.2)=Y(I) /YM
      T50(I.3) = Z(I) /ZM
1     CONTINUE
      RETURN
       END

C
      SUBROUTINE POWERX(XT. VT.XLT.XMT.XMDOTT.TT.GMBOD .TSP.P.PM)
C
C
C       THIS SUBROUTINE COMPUTES THE STATE TRANSITION  MATRICES  P.Q.R
C      AT TIME   T  DURING  POWERED FLIGHT.AND THE FORWARD PROPAGATION
C      MATRIX  PM.
C
      IMPLICIT REAL*8(A-H.K-Z)
      DIMENSION XT(3).VT(3).XLT(3).XTR(3).VTR(3).LTR(3).P(6.6).PM(6.3)
      DO 30 I=1.3
      XTR(I) = XT(I)
      VTR(I) = VT(I)
30    LTR(I) = XLT(I)
      MTR=XMT
      MDOTTR = XMDOTT
      TR = TT
      GMBODY = GMBOD
      T = TSP
      RSQ = ( XTR(1)**2  + XTR(2)**2   + XTR(3)**2 )
      RCUBE = RSQ * DSQRT(RSQ)
      DT = T - TR
      A  =  GMBODY / RCUBE
      B =  A * DT
      C =  B * DT/2.0D00
      D =  3.0D00 * C / RSQ
      E =  DT / RCUBE
      F =  E * DT/ 2.0D00
      G =  3.0D00 * B / RSQ
      H =  MTR + MDOTTR  * DT
      K =  DLOG(H/ MTR)/ MDOTTR
      DO 1  I = 1.3
      DO 2  J = 1.3
      P(I.J) =  D * XTR(I) * XTR(J)
      P(I.J+3) =  0.0D00
      P(I+3.J) =  G * XTR(I) * XTR(J)
      P(I+3.J+3) = 0.0D00
      IF ( I .NE. J ) GO TO 2
      P(I.J) = P(I.J) + 1.0D00 - C
      P(I.J+3) = DT
      P(I+3.J ) = P(I+3.J) - B
      P(I+3.J+3) = 1.0D00
  2   CONTINUE
  1   CONTINUE
      MAG = DSQRT(LTR(1)**2+LTR(2)**2+LTR(3)**2)
      K1 = MDOTTR*K
      XTERM = -(DT-(MTR/MDOTTR+DT)*K1)/MDOTTR
      XTERM1= (2.0D00*DT-(2.0D00*MTR/MDOTTR+DT)*K1)/(MDOTTR*MAG**2)
      DO 10 I=1,3
      DO 10 J=1,3
      PM(I,J)=XTERM1*LTR(I)*LTR(J)
      IF(I.EQ.J) PM(I,J) = PM(I,J)+XTERM
   10 CONTINUE
      XTERM1 = -(K1/MAG-MDOTTR*DT/(MAG*H))/(MDOTTR*MAG)
      DO 20 I=4,6
      DO 20 J=1,3
      PM(I,J) = XTERM1*LTR(I-3)*LTR(J)
      IF(I-3.EQ.J) PM(I,J) = PM(I,J)+K
   20 CONTINUE
      RETURN
      END

C
      SUBROUTINE MTRPLY (A,B,C,NRA, NCA, NCB, NA,NB,NC)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION A(NA,1), B(NB,1), C(NC,1)
      DO 1 I=1,NRA
      DO 1 J=1,NCB
      C(I,J) = 0.D0
      DO 1 K=1,NCA
      C(I,J) = C(I,J) + A(I,K)*B(K,J)
    1 CONTINUE
      RETURN
       END

C
      SUBROUTINE TWOBDY(S0,TAU,MU,PSI,S,P,PI,PMU,P0MU,ACC,ACC0,R,R0)
C
C
C GENERAL SOLUTION OF TWO BODY PROBLEM WITH PARTIAL DERIVATIVES
C FORTRAN 4 DOUBLE PRECISION SUBROUTINE FOR IBM 7094 WITH IBSYS SYSTEM
C SEE APRIL 1965 ASTRONOMICAL JOURNAL FOR FORMULATION BY W. H. GOODYEAR
C
C CALLING SEQUENCE IS AS FOLLOWS
C     CALL TWOBDY(S0,TAU,MU,PSI,S,P,PI,PMU,P0MU,ACC,ACC0,R,R0)
C
C DOUBLE PRECISION QUANTITIES IN CALLING SEQUENCE ARE AS FOLLOWS
      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION S0(6),TAU,MU,PSI
     1,S(6),P(6,6),PI(6,6),PMU(6),P0MU(6),ACC(3),ACC0(3),R,R0
C
C               INPUTS
C S0(1),S0(2),S0(3)=X0,Y0,Z0=POSITION COMPONENTS AT REFERENCE TIME T0
C S0(4),S0(5),S0(6)=XD0,YD0,ZD0=VELOCITY COMPONENTS AT REFERENCE TIME T0
C TAU=TIME INTERVAL (T-T0) FROM REFERENCE TIME T0 TO SOLUTION TIME T
C MU=CONSTANT IN DIFFERENTIAL EQUATIONS (XDD,YDD,ZDD)=-MU*(X,Y,Z)/(R**3)
C PSI=APPROXIMATION FOR FINAL SOLUTION PSI OF KEPLER'S EQUATION
C
C               OUTPUTS
C PSI=GENERALIZED ECCENTRIC ANOMALY=SOLUTION OF KEPLERS EQUATION
C S(1),S(2),S(3)=X,Y,Z=POSITION COMPONENTS AT SOLUTION TIME T=T0+TAU
C S(4),S(5),S(6)=XD,YD,ZD=VELOCITY COMPONENTS AT SOLUTION TIME T=T0+TAU
C P(I,J)=PARTIAL DERIVATIVE DS(I)/DS0(J) OF S(I) WITH RESPECT TO S0(J)
C PI(I,J)=PARTIAL DS0(I)/DS(J) WITH ROLES OF T0 AND T REVERSED
C PMU(I)=PARTIAL DS(I)/DMU OF S(I) WITH RESPECT TO MU
C P0MU(I)=PARTIAL DS0(I)/DMU WITH ROLES OF T0 AND T REVERSED
C ACC(I)=-MU*S(I)/(R**3)=ACCELERATION COMPONENT AT SOLUTION TIME T
C ACC0(I)=-MU*S0(I)/(R0**3)=ACCELERATION COMPONENT AT REFERENCE TIME T0
C R=RADIUS AT TIME T=SQUARE ROOT OF(X**2+Y**2+Z**2)
C R0=RADIUS AT TIME T0=SQUARE ROOT OF(X0**2+Y0**2+Z0**2)
C
C ADDITIONAL DOUBLE PRECISION QUANTITIES FOR COMPUTATION
     2,SIG0,ALPHA,PSIN,PSIP,A,AP,C0,C1,C2,C3,C4,C5X3,S1,S2,S3,DTAU,DTAUN
     3,DTAUP,U,FM1,G,FD,GDM1
C
C START OF INITIAL COMPUTATIONS
C COMPUTE RADIUS R0=SQUARE ROOT OF(X0**2+Y0**2+Z0**2)
      S1=DMAX1(DABS(S0(1)),DABS(S0(2)),DABS(S0(3)))
      S2=(S0(1)/S1)**2+(S0(2)/S1)**2+(S0(3)/S1)**2
      R0=2.D0
   10 R=R0
      R0=(R+S2/R)*.5D0
      IF(R0.LT.R) GO TO 10
      R0=R0*S1
C COMPUTE OTHER PARAMETERS
```

```
      SIGO=SO(1)*SO(4)+SO(2)*SO(5)+SO(3)*SO(6)
      ALPHA=SO(4)**2+SO(5)**2+SO(6)**2-2.DO*MU/RO
C INITIALIZE SERIES MOD COUNT M TO ZERO
      M=0
C INITIALIZE BOUNDS PSIN AND PSIP FOR PSI OR SET PSI=0 IF TAU=0
      IF(TAU) 20,30,40
   20 PSIN=-1.D+38
      PSIP=0.DO
      DTAUN=PSIN
      DTAUP=-TAU
      GO TO 50
   30 PSI=0.DO
      GO TO 100
   40 PSIN=0.DO
      PSIP=+1.D+38
      DTAUN=-TAU
      DTAUP=PSIP
C USE APPROXIMATION FOR PSI IF IT IS BETWEEN BOUNDS PSIN AND PSIP
   50 IF(PSI.GT.PSIN.AND.PSI.LT.PSIP) GO TO 100
C TRY NEWTON'S METHOD FOR INITIAL PSI SET EQUAL TO ZERO
      PSI=TAU/RO
C SET PSI=TAU IF NEWTON'S METHOD FAILS
      IF(PSI.LE.PSIN.OR.PSI.GE.PSIP) PSI=TAU
C END OF INITIAL COMPUTATIONS
C
C BEGINNING OF LOOP FOR SOLVING KEPLER'S EQUATION
C BEGINNING OF SERIES SUMMATION
C COMPUTE ARGUMENT A IN REDUCED SERIES OBTAINED BY FACTORING OUT PSI'S
  100 A=ALPHA*PSI*PSI
      IF(DABS(A).LE.1.DO) GO TO 120
C SAVE A IN AP AND MOD A IF IT EXCEEDS UNITY IN MAGNITUDE
      AP=A
  110 M=M+1
      A=A*.25DO
      IF(DABS(A).GT.1.DO) GO TO 110
C SUM SERIES C5X3=3*S5/PSI**5 AND C4=S4/PSI**4
  120 C5X3=(1.DO+(1.DO+(1.DO+(1.DO+(1.DO+(1.DO+(1.DO+A/342.DO)*A/272.DO)
     1*A/210.DO)*A/156.DO)       *A/110.DO)*A/72.DO)*A/42.DO)/40.DO
      C4  =(1.DO+(1.DO+(1.DO+(1.DO+(1.DO+(1.DO+(1.DO+A/306.DO)*A/240.DO)
     1*A/182.DO)*A/132.DO)       *A/90.DO)*A/56.DO)*A/30.DO)/24.DO
C COMPUTE SERIES C3=S3/PSI**3,C2=S2/PSI**2,C1=S1/PSI,CO=SO
      C3=(.5DO+A*C5X3)/3.DO
      C2= .5DO+A*C4
      C1= 1.DO+A*C3
      CO= 1.DO+A*C2
      IF(M.LE.O) GO TO 140
C DEMOD SERIES CO AND C1 IF NECESSARY WITH DOUBLE ANGLE FORMULAS
  130 C1=C1*CO
      CO=2.DO*CO*CO-1.DO
      M=M-1
      IF(M.GT.O) GO TO 130
C DETERMINE C2,C3,C4,C5X3 FROM CO,C1,AP IF DEMOD REQUIRED
      C2=(CO-1.DO)/AP
      C3=(C1-1.DO)/AP
      C4=(C2-.5DO)/AP
      C5X3=(3.DO*C3-.5DO)/AP
C COMPUTE SERIES S1,S2,S3 FROM C1,C2,C3
  140 S1=C1*PSI
      S2=C2*PSI*PSI
      S3=C3*PSI*PSI*PSI
C END OF SERIES SUMMATION
C COMPUTE RESIDUAL DTAU AND SLOPE R FOR KEPLER'S EQUATION
      G=RO*S1+SIGO*S2
      DTAU=(G+MU*S3)-TAU
      R=DABS(RO*CO+(SIGO*S1+MU*S2))
      IF(DTAU) 200,300,210
C RESET BOUND
  200 PSIN=PSI
      DTAUN=DTAU
      GO TO 220
  210 PSIP=PSI
      DTAUP=DTAU
C TRY NEWTON'S METHOD AND INITIALIZE SELECTOR N
  220 PSI=PSI-DTAU/R
      N=0
C ACCEPT PSI IF IT IS BETWEEN BOUNDS PSIN AND PSIP
  230 IF(PSI.GT.PSIN.AND.PSI.LT.PSIP). GO TO 100
C SELECT ALTERNATE METHOD OF COMPUTING PSI OR STOP ITERATIONS
      N=N+1
      GO TO (1,2,3,4,300),N
C TRY INCREMENTING BOUND WITH DTAU NEAREST ZERO BY THE RATIO 4*DTAU/TAU ,
    1 IF(DABS(DTAUN).LT.DABS(DTAUP)) PSI=PSIN*(1.D0-(4.D0*DTAUN)/TAU)
      IF(DABS(DTAUP).LT.DABS(DTAUN)) PSI=PSIP*(1.D0-(4.D0*DTAUP)/TAU)
      GO TO 230
C TRY DOUBLING BOUND CLOSEST TO ZERO
    2 IF(TAU.GT.0.D0) PSI=PSIN+PSIN
      IF(TAU.LT.0.D0) PSI=PSIP+PSIP
      GO TO 230
C TRY INTERPOLATION BETWEEN BOUNDS
    3 PSI=PSIN+(PSIP-PSIN)*(-DTAUN/(DTAUP-DTAUN))
      GO TO 230
C TRY HALVING BETWEEN BOUNDS
    4 PSI=PSIN+(PSIP-PSIN)*.5D0
      GO TO 230
C END OF LOOP FOR SOLVING KEPLER'S EQUATION
C
C COMPUTE REMAINING THREE OF FOUR FUNCTIONS FM1,G,FD,GDM1
  300 FM1=-MU*S2/R0
      FD=-MU*S1/R0/R
      GDM1=-MU*S2/R
C COMPUTE COORDINATES AT SOLUTION TIME T=T0+TAU
      DO 310 I=1,3
      S(I)=S0(I)+(FM1*S0(I)+G*S0(I+3))
      S(I+3)=(FD*S0(I)+GDM1*S0(I+3))+S0(I+3)
C COMPUTE ACCELERATIONS
      ACC(I)=-MU*S(I)/R/R/R
  310 ACC0(I)=-MU*S0(I)/R0/R0/R0
C END OF COMPUTATION FOR COORDINATES AND ACCELERATIONS
C
C COMPUTATION OF PARTIAL DERIVATIVES
C COMPUTE COEFFICIENTS FOR STATE PARTIALS
      U= S2*TAU+MU*(C4-C5X3)*PSI*PSI*PSI*PSI*PSI
      P(1,1)=-(FD*S1+FM1/R0)/R0
      P(1,2)=-FD*S2
      P(2,1)= FM1*S1/R0
      P(2,2)= FM1*S2
      P(1,3)= P(1,2)
      P(1,4)=-GDM1*S2
      P(2,3)= P(2,2)
      P(2,4)= G*S2
      P(3,1)=-FD*(C0/R0/R+1.D0/R/R+1.D0/R0/R0)
      P(3,2)=-(FD*S1+GDM1/R)/R
      P(4,1)=-P(1,1)
      P(4,2)=-P(1,2)
      P(3,3)= P(3,2)
      P(3,4)=-GDM1*S1/R
      P(4,3)=-P(1,2)
      P(4,4)=-P(1,4)
C COMPUTE COEFFICIENTS FOR MU PARTIALS
      P(1,5)=-S1/R0/R
      P(2,5)= S2/R0
      P(3,5)= U/R0-S3
      P(1,6)=-P(1,5)
      P(2,6)= S2/R
      P(3,6)=-U/R+S3
      DO 400 I=1,3
C COMPUTE MU PARTIALS
      PMU(I)=-S(I)*P(2,5)+S(I+3)*P(3,5)
      PMU(I+3)= S(I)*P(1,5)+S(I+3)*P(2,5)+ACC(I)*P(3,5)
      P0MU(I)=-S0(I)*P(2,6)+S0(I+3)*P(3,6)
      P0MU(I+3)= S0(I)*P(1,6)+S0(I+3)*P(2,6)+ACC0(I)*P(3,6)
C MATRIX ACCUMULATIONS FOR STATE PARTIALS
      DO 400 J=1,4
      PI(J,I)= P(J,1)*S0(I)+P(J,2)*S0(I+3)
  400 PI(J,I+3)= P(J,3)*S0(I)+P(J,4)*S0(I+3)
      DO 410 I=1,3
      DO 420 J=1,3
      P(I,J)     =S(I)*PI(1,J)  +S(I+3)*PI(2,J)  +U*S(I+3)*ACC0(J)
      P(I,J+3)   =S(I)*PI(1,J+3)+S(I+3)*PI(2,J+3)-U*S(I+3)*S0(J+3)
      P(I+3,J)   =S(I)*PI(3,J)  +S(I+3)*PI(4,J)  +U*ACC(I)*ACC0(J)
  420 P(I+3,J+3)=S(I)*PI(3,J+3)+S(I+3)*PI(4,J+3)-U*ACC(I)*S0(J+3)
      P(I,I)     =P(I,I)    +FM1+1.D0
      P(I,I+3)   =P(I,I+3)  +G
      P(I+3,I)   =P(I+3,I)  +FD
  410 P(I+3,I+3)=P(I+3,I+3)+GDM1+1.D0
C TRANSPOSITIONS FOR INVERSE STATE PARTIALS
      DO 430 I=1,3
      DO 430 J=1,3
      PI(J+3,I+3)= P(I,J)
      PI(J+3,I)   =-P(I+3,J)
      PI(J,I+3)   =-P(I,J+3)
  430 PI(J,I)     = P(I+3,J+3)
C END OF COMPUTATION FOR PARTIAL DERIVATIVES
C
C END OF PROGRAM - ALL OUTPUTS HAVE BEEN COMPUTED
      RETURN
      END

C
      SUBROUTINE CONVET (P,R,V,K,PQ)
C
C         K=1    NO TRANSFORMATION
C         K=2    P TRANSFORMED FROM LOCAL TANGENT TO EQUATOR 50
C         K=0    P TRANSFORMED FROM EQUATOR 50 TO LOCAL TANGENT
C         Q IS THE TRANSFORMATION
      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION P(6,6), Q(6,6), DUM, PQ(6,6)
      DIMENSION R(3), V(3), RXV(3), RXVXR(3)
      IF (K .EQ. 1) RETURN
      CALL CROSS (R,V,RXV)
      CALL CROSS (RXV, R, RXVXR)
      RN = FNORM(R)
      RXVN = FNORM(RXV)
      RXVXRN = FNORM(RXVXR)
      DO 1 I=1,3
      Q(I,1) = RXVXR(I) /RXVXRN
      Q(I,2) = RXV(I) /RXVN
      Q(I,3) = R(I) /RN
      DO 2 J=1,3
      Q(I+3 ,J+3) = Q(I,J)
      Q(I+3, J) = 0.D0
      Q(I, J+3) = 0.D0
2     CONTINUE
1     CONTINUE
      IF (K .EQ. 2) GO TO 3
      DO 4 I=1,6
      DO 5 J=1,I
      DUM = Q(I,J)
      Q(I,J) = Q(J,I)
5     Q(J,I) = DUM
4     CONTINUE
3     CALL MTRX (Q, P, PQ, 6, 6,-1)
      RETURN
       END

C
      SUBROUTINE MTRX(A,B,Q,NR,NC,M)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION A,B,C,D,Q
      DIMENSION A(6,6),B(NR,NC),C(6,6),D(6),Q(6,6)
C     M=0  GIVES Q=AB
C     M=-1 GIVES Q=ABATRANSPOSE
      DO 7 I=1,6
      DO 1 J=1,NC
      D(J)=0.D0
      DO 1 K=1,NR
1     D(J)=D(J)+A(I,K)*B(K,J)
      IF(M) 4,2,4
2     CONTINUE
      DO 3 J=1,NC
3     C(I,J)=D(J)
      GO TO 7
4     DO 5 J=1,6
      C(I,J)=0.D0
      DO 5 K=1,NC
5     C(I,J)=C(I,J)+D(K)*A(J,K)
7     CONTINUE
      DO 8 I=1,6
      DO 8 J=1,6
8     Q(I,J)=C(I,J)
      RETURN
      END

C
      SUBROUTINE TAB1(A,S,NOVAR,UBO,FREQ,PCT,STATS,NO,NV)
```

```
C
      REAL*8 Z(4)
      DIMENSION A(1),S(1),UBO(1),FREQ(1),PCT(1),STATS(1)
      DIMENSION WBO(3)
      DO 5 I = 1,3
    5 WBO(I) = UBO(I)
      VMIN = 1.0E75
      VMAX = -1.0E75
      IJ = NO*(NOVAR-1)
      DO 30 J = 1,NO
      IJ = IJ+1
      IF (S(J)) 10,30,10
   10 IF (A(IJ)-VMIN) 15,20,20
   15 VMIN = A(IJ)
   20 IF (A(IJ)-VMAX) 30,30,25
   25 VMAX = A(IJ)
   30 CONTINUE
      STATS(4) = VMIN
      STATS(5) = VMAX
      IF (UBO(1)-UBO(3)) 40,35,40
   35 UBO(1) = VMIN
      UBO(3) = VMAX
   40 INN = UBO(2)
      DO 45 I = 1,INN
      FREQ(I) = 0.0
   45 PCT(I) = 0.0
      DO 50 I = 1,3
      Z(I) = 0.D0
   50 STATS(I) = 0.0
      SINT = ABS((UBO(3)-UBO(1))/(UBO(2)-2.0))
      SCNT = 0.0
      IJ = NO*(NOVAR-1)
      DO 75 J = 1,NO
      IJ = IJ+1
      IF (S(J)) 55,75,55
   55 SCNT = SCNT+1.0
      STATS(1) = STATS(1)+A(IJ)
      ST ATS(3) = STATS(3)+A(IJ)*A(IJ)
      Z(4) = A(IJ)
      Z(1) = Z(1)+Z(4)
      Z(3) = Z(3)+Z(4)**2
      TEMP = UBO(1)-SINT
      INTX = INN-1
      DO 60 I = 1,INTX
      TEMP = TEMP+SINT
      IF (A(IJ)-TEMP) 70,60,60
   60 CONTINUE
      IF (A(IJ)-TEMP) 75,65,65
   65 FREQ(INN) = FREQ(INN)+1.0
      GO TO 75
   70 FREQ(I) = FREQ(I)+1.0
   75 CONTINUE
      STATS(1) = Z(1)
      STATS(3) = Z(3)
      DO 80 I = 1,INN
   80 PCT(I) = FREQ(I)*100.0/SCNT
      IF (SCNT-1.0) 85,85,90
   85 STATS(2) = 0.0
      STATS(3) = 0.0
      GO TO 95
   90 STATS(2) = STATS(1)/SCNT
      STATS(3) = SQRT(ABS((STATS(3)-STATS(1)*STATS(1)/SCNT)/(SCNT-1.0)))
   95 DO 100 I = 1,3
  100 UBO(I) = WBO(I)
      RETURN
      END

C
      REAL FUNCTION DOT*8(X,Y)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(3),Y(3)
      DOT = X(1)*Y(1) + X(2)*Y(2) + X(3)*Y(3)
      RETURN
      END

C
```

```
      REAL FUNCTION ARKTNS*8 (N,X,Y)
C
      IMPLICIT REAL*8(A-H,O-Z)
C     COMPUTES 4-QUADRANT ARCTANGENT OF Y/X IN RADIANS
C     N=360  ANGLE LIES IN RANGE (0,360) DEG
C     N=180  ANGLE LIES IN RANGE (-180,180) DEG
      TPI=6.283185307179586D0
      XA=DABS(X)
      YA=DABS(Y)
      IF(XA-YA)1,1,2
    1 Z=X/YA
      GO TO 3
    2 Z=Y/XA
      YA=XA
    3 D=DSQRT(1.0D0+Z*Z)
      YA=YA*D+X
      IF(YA)4,4,5
    4 ARKTNS=TPI/2.0D0
      GO TO 6
    5 ARKTNS=2.0D0*DATAN(Y/YA)
    6 IF(N-180)7,9,7
    7 IF(ARKTNS)8,9,9
    8 ARKTNS=ARKTNS+TPI
    9 RETURN
      END

C
      REAL FUNCTION FNORM*8(X)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(3)
    1 FNORM =DSQRT(X(1)**2+X(2)**2+X(3)**2)
    3 RETURN
      END

C
      SUBROUTINE CROSS(A,B,C)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(3),B(3),C(3)
C     DIMENSION A(3),B(3),C(3)
      C(1)=A(2)*B(3)-A(3)*B(2)
      C(2)=A(3)*B(1)-A(1)*B(3)
      C(3)=A(1)*B(2)-A(2)*B(1)
      RETURN
      END

C
      SUBROUTINE ORB(X,DX,U,OE)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION OE(6)
      DIMENSION X(3),DX(3),B(3)
C THE FOLLOWING STATEMENT(S) HAVE BEEN MANUFACTURED BY THE TRANSLATOR TO
C   COMPENSATE FOR THE FACT THAT EQUIVALENCE DOES NOT REORDER COMMON---
C     DIMENSION X(3),DX(3),B(3)
      CALL CROSS(X,DX,B)
      R2 = DOT(X,X)
      R  = DSQRT(R2)
      V2 = DOT(DX,DX)
      B2 = DOT(B,B)
      BB = DSQRT(B2)
      A=DOT(X,DX)/U
      P  = B2/U
      C3 = V2-2.0D0*U/R
      SMA=-U/C3
      ECC=DSQRT(1.0D0+C3*P/U)
      OINC= ARKTNS(180,B(3),DSQRT(B(1)**2+B(2)**2))
      OMG= ARKTNS(360,-B(2),B(1))
      RCA=P/(1.0D0+ECC)
      THT=ARKTNS(360,(P-R),BB*A)
      BET= ARKTNS(360,X(2)*B(1)-X(1)*B(2),X(3)*BB)
      BEP= BET-THT
      IF(BEP) 2,3,3
    2 BEP=BEP+6.2831853D0
    3 CONTINUE
      RTD=57.29577951D0
      OINC=OINC*RTD
      OMG=OMG*RTD
      BEP=BEP*RTD
      PERV=DSQRT(C3+2.0D0*U/RCA)
      VIMP=PERV-DSQRT(U/RCA)
      CTAS=(P/R-1.0D0)/ECC
      IF(CTAS.LT.1.D0.OR.(CTAS-1.D0).GT.1.D-6) GO TO 200
      STAS=0.D0
      GO TO 201
  200 STAS=DSQRT(1.0D0-CTAS*CTAS)
      STAS=DSIGN(STAS,A)
  201 CONTINUE
      THE=ARKTNS(180,CTAS,STAS)
      THET=THE*RTD
      CALL TCONIC(U,ECC,SMA,P,THE,TPER,FAC)
      TPER=TPER/86400.0D0
      IF(SMA)10,10,11
   10 CONTINUE
      GO TO 12
   11 F1=A*U/DSQRT(U*SMA)
      F2=1.0D0-R/SMA
      SINE=F1/ECC
      COSE=F2/ECC
      E=ARKTNS(360,COSE,SINE)
      XMAN=(E-ECC*SINE)*RTD
   12 CONTINUE
      OE(1) = ECC
      OE(2) = SMA
      OE(3) = OINC
      OE(4) = OMG
      OE(5) = BEP
      OE(6) = THET
      RETURN
      END

C
      SUBROUTINE TCONIC(U,EC,A,SLR,TA2,T,FAC)
C
      IMPLICIT REAL*8(A-H,O-Z)
      TANG(Q000FL)=DSIN(Q000FL)/DCOS(Q000FL)
      AB=DABS(A)
      FAC=AB*DSQRT(AB/U)
      ECA=(1.0D0-EC)/(1.0D0+EC)
      ABE=DSQRT(DABS(ECA))
      THE=TANG(.5D0*TA2)
      IF(ABE-.00005D0)11,11,12
   12 CONTINUE
      ECA=2.0D0*DATAN(ABE*THE)
      IF(A)14,11,13
   13 T=FAC*(ECA-EC*DSIN(ECA))
      GO TO 16
   14 ANG=.78539816 3D0+.5D0*ECA
      T=FAC*(EC*TANG(ECA)-DLOG(TANG(ANG)))
      GO TO 16
   11 FAC=DSQRT(SLR**3/U)*2.0D0/((1.0D0+EC)**2)
      EC1=ECA*THE**2
      T=FAC*(THE+THE**3*((1.0D0-2.0D0*ECA)/3.0D0-(2.0D0-3.0D0*ECA)*EC1/5
     1.0D0+(3.0D0-4.0D0*ECA)*EC1**2/7.0D0-(4.0D0-5.0D0*ECA)*EC1**3/9.0D0
     *))
   16 CONTINUE
      RETURN
      END

      SUBROUTINE HISTO (TITLE,NAME,P,X,XM,XS,NP,XNOM,NS)
      DIMENSION TITLE(20),NAME(2),P(1),X(1),AL1(50)
      DATA BLANK/' '/
      DATA EYE/'I'/
      WRITE (6,600) TITLE
      WRITE (6,610) NAME,XNOM,XM,XS,NS
      WRITE (6,620) X(1),P(1),P(1)
      PMAX = P(1)
      DO 10 I = 2,NP
      PMAX = AMAX1(PMAX,P(I))
   10 CONTINUE
      DP = .05
      IF (PMAX.GT.2.5) DP = .1
      IF (PMAX.GT.5.0) DP = .2
      IF (PMAX.GT.10.) DP = .5
      IF (PMAX.GT.25.) DP = 1.
      IF (PMAX.GT.50.) DP = 2.
      DO 20 I = 1,50
      AL1(I) = BLANK
   20 CONTINUE
      ALN = DP*50.
      SUM = P(1)
      DO 50 I = 1,50
      DO 30 J = 1,NP
      IF (P(J)+DP/2..GE.ALN) AL1(J) = EYE
   30 CONTINUE
      IP1 = I+1
      SUM = SUM+P(I+1)
      IF (I.GT.NP-2) GO TO 40
      WRITE (6,650) ALN,AL1,IP1,X(I),X(I+1),P(I+1),SUM
      GO TO 48
   40 IF (I.GT.NP-1) GO TO 45
      WRITE (6,660) ALN,AL1,IP1,X(I),P(I+1),SUM
      GO TO 48
   45 WRITE (6,650) ALN,AL1
   48 ALN = ALN-DP
   50 CONTINUE
      WRITE (6,670) (I,I=5,NP,5)
      WRITE (6,680)
      RETURN
  600 FORMAT (1H1,4X,20A4)
  620 FORMAT (/5X,' PCT. ',50X,5X,9X,'INTERVALS',12X,'PCT. FREQ.',5X,
    2 'SUM'/5X,' FREQ.',50X,5X,'  1   MINUS INF. ',1PE12.4,0P2F12.7)
  610 FORMAT (5X,2A4,2X,'NOMINAL=',1PE12.4,2X,'MEAN=',1PE12.4, 2X,
    2 'SIGMA=',1PE12.4,2X,'SAMPLE=',I6)
  650 FORMAT (5X,F5.2,1X,50A1,5X,I3,1X,1P2E12.4,0P2F12.7)
  660 FORMAT (5X,F5.2,1X,50A1,5X,I3,1PE13.4,'   PLUS INF. ',0P2F12.7)
  670 FORMAT (5X,6X,20I5)
  680 FORMAT (5X,6X, 5X,'I N T E R V A L S')
      END

      SUBROUTINE DELVS (A,RA,RP,AS,ES,AI,AIS,APF,XMU,DVS)
      IMPLICIT REAL*8(A-H,O-Z)
      RAS = AS*(1.D0+ES)
      RPS = AS*(1.D0-ES)
      VAS = DSQRT(XMU*(2.D0/RAS-1.D0/AS))
      VPS = DSQRT(XMU*(2.D0/RPS-1.D0/AS))
      ANA = (RA+RPS)/2.D0
      ANP = (RP+RAS)/2.D0
      VA   = DSQRT(XMU*(2.D0/RA-1.D0/A   ))
      VP   = DSQRT(XMU*(2.D0/RP-1.D0/A   ))
      VAN1 = DSQRT(XMU*(2.D0/RA-1.D0/ANA))
      VPN1 = DSQRT(XMU*(2.D0/RP-1.D0/ANP))
      VAN2 = DSQRT(XMU*(2.D0/RPS-1.D0/ANA))
      VPN2 = DSQRT(XMU*(2.D0/RAS-1.D0/ANP))
      DVA1 = DABS(VAN1-VA)
      DVA2 = DABS(VAN2-VPS)
      DVP1 = DABS(VPN1-VP)
      DVP2 = DABS(VPN2-VAS)
      IF (AIS.GE.0.D0) GO TO 100
      DVS = DMIN1(DVA1+DVA2,DVP1+DVP2)
      RETURN
  100 CAPF = DCOS(APF)
      R1 = RP*RA/A  /(1.D0+(RA/A  -1.D0)*CAPF)
      R2 = RP*RA/A  /(1.D0-(RA/A  -1.D0)*CAPF)
      R3 = RPS*RA/ANA/(1.D0+(RA/ANA-1.D0)*CAPF)
      R4 = RPS*RA/ANA/(1.D0-(RA/ANA-1.D0)*CAPF)
      R5 = RP*RAS/ANP/(1.D0+(RAS/ANP-1.D0)*CAPF)
      R6 = RP*RAS/ANP/(1.D0-(RAS/ANP-1.D0)*CAPF)
      R7 = RPS*RAS/AS/(1.D0+(RAS/AS-1.D0)*CAPF)
      R8 = RPS*RAS/AS/(1.D0-(RAS/AS-1.D0)*CAPF)
      V1 = DSQRT (XMU*(2.D0/R1-1.D0/A   ))
      V2 = DSQRT (XMU*(2.D0/R2-1.D0/A   ))
      V3 = DSQRT (XMU*(2.D0/R3-1.D0/ANA))
      V4 = DSQRT (XMU*(2.D0/R4-1.D0/ANA))
      V5 = DSQRT (XMU*(2.D0/R5-1.D0/ANP))
      V6 = DSQRT (XMU*(2.D0/R6-1.D0/ANP))
      V7 = DSQRT (XMU*(2.D0/R7-1.D0/AS))
      V8 = DSQRT (XMU*(2.D0/R8-1.D0/AS))
      DI = DABS(AI-AIS)
      DVS = DMIN1 (DI*V1+DVA1+DVA2, DI*V2+DVA1+DVA2, DI*V1+DVP1+DVP2,
    2             DI*V2+DVP1+DVP2, DI*V3+DVA1+DVA2, DI*V4+DVA1+DVA2,
    3             DI*V5+DVP1+DVP2, DI*V6+DVP1+DVP2, DI*V7+DVA1+DVA2,
    4             DI*V8+DVA1+DVA2, DI*V7+DVP1+DVP2, DI*V8+DVP1+DVP2)
      RETURN
      END
```